TRAITE DE COOPERATION EN MATIERE DE BREVETS

Expéditeur: le BUREAU INTERNATIONAL

PCT

NOTIFICATION D'ELECTION

(règle 61.2 du PCT)

Destinataire:

Assistant Commissioner for Patents
United States Patent and Trademark Office
Box PCT
Washington, D.C.20231
ETATS-UNIS D'AMERIQUE

en sa qualité d'office élu

Date d'expédition (jour/mois/année) 07 juillet 2000 (07.07.00)	
Demande internationale no PCT/FR99/02790	**Référence du dossier du déposant ou du mandataire** BRL 8 PCT
Date du dépôt international (jour/mois/année) 15 novembre 1999 (15.11.99)	**Date de priorité** (jour/mois/année) 19 novembre 1998 (19.11.98)
Déposant BARLIER, Claude	

1. L'office désigné est avisé de son élection qui a été faite:

[X] dans la demande d'examen préliminaire international présentée à l'administration chargée de l'examen préliminaire international le:

15 juin 2000 (15.06.00)

[ ] dans une déclaration visant une élection ultérieure déposée auprès du Bureau international le:

2. L'élection [X] a été faite
[ ] n'a pas été faite

avant l'expiration d'un délai de 19 mois à compter de la date de priorité ou, lorsque la règle 32 s'applique, dans le délai visé à la règle 32.2b).

Bureau international de l'OMPI **34, chemin des Colombettes** **1211 Genève 20, Suisse** no de télécopieur: (41-22) 740.14.35	Fonctionnaire autorisé Diana Nissen no de téléphone: (41-22) 338.83.38

Express Mail No. ET169352964US

PCT

REQUEST

The undersigned requests that the present international application be processed according to the Patent Cooperation Treaty.

For receiving Office use only

International Application No.

International Filing Date 09/856311

Name of receiving Office and "PCT International Application"

Applicant's or agent's file reference *(if desired) (12 characters maximum)* BRL 8 PCT

Box No. I TITLE OF INVENTION METHOD FOR MAKING MECHANICAL PARTS BY DECOMPOSITION INTO LAYERS

Box No. II APPLICANT

Name and address: *(Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country. The country of the address indicated in this Box is the applicant's State (that is, country) of residence if no State of residence is indicated below.)*

C.I.R.T.E.S. (Partnership Law 1901)
(Centre d'Ingéniérie de Recherche et de Transfert de l'ESSTIN à Saint-Dié)
29 Bis rue d'Hellieule
88100 SAINT DIE

☐ This person is also inventor.

Telephone No.

Facsimile No.

Teleprinter No.

State *(that is, country)* of nationality: FRANCE

State *(that is, country)* of residence: FRANCE

This person is applicant for the purposes of: ☐ all designated States ☒ all designated States except the United States of America ☐ the United States of America only ☐ the States indicated in the Supplemental Box

Box No. III FURTHER APPLICANT(S) AND/OR (FURTHER) INVENTOR(S)

Name and address: *(Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country. The country of the address indicated in this Box is the applicant's State (that is, country) of residence if no State of residence is indicated below.)*

BARLIER, Claude
67 Chemin de la Roche
88100 COINCHES

This person is:
☐ applicant only
☒ applicant and inventor
☐ inventor only *(If this check-box is marked, do not fill in below.)*

State *(that is, country)* of nationality: FRANCE

State *(that is, country)* of residence: FRANCE

This person is applicant for the purposes of: ☐ all designated States ☐ all designated States except the United States of America ☒ the United States of America only ☐ the States indicated in the Supplemental Box

☐ Further applicants and/or (further) inventors are indicated on a continuation sheet.

Box No. IV AGENT OR COMMON REPRESENTATIVE; OR ADDRESS FOR CORRESPONDENCE

The person identified below is hereby/has been appointed to act on behalf of the applicant(s) before the competent International Authorities as: ☒ agent ☐ common representative

Name and address: *(Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country.)*

POUPON, Michel
Cabinet Michel POUPON
3 rue Ferdinand Brunot
88026 EPINAL CEDEX
FRANCE

Telephone No. 03.29.64.05.93

Facsimile No. 03.29.64.17.33

Teleprinter No.

☐ **Address for correspondence:** Mark this check-box where no agent or common representative is/has been appointed and the space above is used instead to indicate a special address to which correspondence should be sent.

Box No.V DESIGNATION OF STATES

The following designations are hereby made under Rule 4.9(a) *(mark the applicable check-boxes; at least one must be marked)*:

Regional Patent

☒ **AP** **ARIPO Patent:** **GH** Ghana, **GM** Gambia, **KE** Kenya, **LS** Lesotho, **MW** Malawi, **SD** Sudan, **SL** Sierra Leone, **SZ** Swaziland, **UG** Uganda, **ZW** Zimbabwe, and any other State which is a Contracting State of the Harare Protocol and of the PCT

☒ **EA** **Eurasian Patent:** **AM** Armenia, **AZ** Azerbaijan, **BY** Belarus, **KG** Kyrgyzstan, **KZ** Kazakhstan, **MD** Republic of Moldova, **RU** Russian Federation, **TJ** Tajikistan, **TM** Turkmenistan, and any other State which is a Contracting State of the Eurasian Patent Convention and of the PCT

☒ **EP** **European Patent:** **AT** Austria, **BE** Belgium, **CH and LI** Switzerland and Liechtenstein, **CY** Cyprus, **DE** Germany, **DK** Denmark, **ES** Spain, **FI** Finland, **FR** France, **GB** United Kingdom, **GR** Greece, **IE** Ireland, **IT** Italy, **LU** Luxembourg, **MC** Monaco, **NL** Netherlands, **PT** Portugal, **SE** Sweden, and any other State which is a Contracting State of the European Patent Convention and of the PCT

☒ **OA** **OAPI Patent:** **BF** Burkina Faso, **BJ** Benin, **CF** Central African Republic, **CG** Congo, **CI** Côte d'Ivoire, **CM** Cameroon, **GA** Gabon, **GN** Guinea, **GW** Guinea-Bissau, **ML** Mali, **MR** Mauritania, **NE** Niger, **SN** Senegal, **TD** Chad, **TG** Togo, and any other State which is a member State of OAPI and a Contracting State of the PCT *(if other kind of protection or treatment desired, specify on dotted line)* .....

National Patent *(if other kind of protection or treatment desired, specify on dotted line)*:

☒ **AE** United Arab Emirates
☒ **AL** Albania
☒ **AM** Armenia
☒ **AT** Austria
☒ **AU** Australia
☒ **AZ** Azerbaijan
☒ **BA** Bosnia and Herzegovina
☒ **BB** Barbados
☒ **BG** Bulgaria
☒ **BR** Brazil
☒ **BY** Belarus
☒ **CA** Canada
☒ **CH and LI** Switzerland and Liechtenstein
☒ **CN** China
☒ **CU** Cuba
☒ **CZ** Czech Republic
☒ **DE** Germany
☒ **DK** Denmark
☒ **EE** Estonia
☒ **ES** Spain
☒ **FI** Finland
☒ **GB** United Kingdom
☒ **GD** Grenada
☒ **GE** Georgia
☒ **GH** Ghana
☒ **GM** Gambia
☒ **HR** Croatia
☒ **HU** Hungary
☒ **ID** Indonesia
☒ **IL** Israel
☒ **IN** India
☒ **IS** Iceland
☒ **JP** Japan
☒ **KE** Kenya
☒ **KG** Kyrgyzstan
☒ **KP** Democratic People's Republic of Korea
☒ **KR** Republic of Korea
☒ **KZ** Kazakhstan
☒ **LC** Saint Lucia
☒ **LK** Sri Lanka
☒ **LR** Liberia
☒ **LS** Lesotho
☒ **LT** Lithuania
☒ **LU** Luxembourg
☒ **LV** Latvia
☒ **MD** Republic of Moldova
☒ **MG** Madagascar
☒ **MK** The former Yugoslav Republic of Macedonia
☒ **MN** Mongolia
☒ **MW** Malawi
☒ **MX** Mexico
☒ **NO** Norway
☒ **NZ** New Zealand
☒ **PL** Poland
☒ **PT** Portugal
☒ **RO** Romania
☒ **RU** Russian Federation
☒ **SD** Sudan
☒ **SE** Sweden
☒ **SG** Singapore
☒ **SI** Slovenia
☒ **SK** Slovakia
☒ **SL** Sierra Leone
☒ **TJ** Tajikistan
☒ **TM** Turkmenistan
☒ **TR** Turkey
☒ **TT** Trinidad and Tobago
☒ **UA** Ukraine
☒ **UG** Uganda
☒ **US** United States of America
☒ **UZ** Uzbekistan
☒ **VN** Viet Nam
☒ **YU** Yugoslavia
☒ **ZA** South Africa
☒ **ZW** Zimbabwe

Check-boxes reserved for designating States which have become party to the PCT after issuance of this sheet:

☒ CR Costa Rica
☒ DM Dominica
X Morocco (MA)

Precautionary Designation Statement: In addition to the designations made above, the applicant also makes under Rule 4.9(b) all other designations which would be permitted under the PCT except any designation(s) indicated in the Supplemental Box as being excluded from the scope of this statement. The applicant declares that those additional designations are subject to confirmation and that any designation which is not confirmed before the expiration of 15 months from the priority date is to be regarded as withdrawn by the applicant at the expiration of that time limit. *(Confirmation of a designation consists of the filing of a notice specifying that designation and the payment of the designation and confirmation fees. Confirmation must reach the receiving Office within the 15-month time limit.)*

Supplemental B x *If the Supplemental Box is not used, this sheet should not be included in the request.*

*1. If, in any of the Boxes, **the space is insufficient** to furnish all the information: in such case, write "Continuation of Box No. ..." [indicate the number of the Box] and furnish the information in the same manner as required according to the captions of the Box in which the space was insufficient, in particular:*

*(i) **if more than two persons are involved as applicants and/or inventors** and no "continuation sheet" is available: in such case, write "Continuation of Box No. III" and indicate for each additional person the same type of information as required in Box No. III. The country of the address indicated in this Box is the applicant's State (that is, country) of residence if no State of residence is indicated below;*

*(ii) if, in Box No. II or in any of the sub-boxes of Box No. III, the indication **"the States indicated in the Supplemental Box"** is checked: in such case, write "Continuation of Box No. II" or "Continuation of Box No. III" or "Continuation of Boxes No. II and No. III" (as the case may be), indicate the name of the applicant(s) involved and, next to (each) such name, the State(s) (and/or, where applicable, ARIPO, Eurasian, European or OAPI patent) for the purposes of which the named person is applicant;*

*(iii) if, in Box No. II or in any of the sub-boxes of Box No. III, **the inventor or the inventor/applicant is not inventor for the purposes of all designated States or for the purposes of the United States of America**: in such case, write "Continuation of Box No. II" or "Continuation of Box No. III" or "Continuation of Boxes No. II and No. III" (as the case may be), indicate the name of the inventor(s) and, next to (each) such name, the State(s) (and/or, where applicable, ARIPO, Eurasian, European or OAPI patent) for the purposes of which the named person is inventor;*

*(iv) if, in addition to the agent(s) indicated in Box No. IV, there are **further agents**: in such case, write "Continuation of Box No. IV" and indicate for each further agent the same type of information as required in Box No. IV;*

*(v) if, in Box No. V, the name of any State (or OAPI) is accompanied by the indication "**patent of addition**," or "**certificate of addition**," or if, in Box No. V, the name of the United States of America is accompanied by an indication "**continuation**" or "**continuation-in-part**": in such case, write "Continuation of Box No. V" and the name of each State involved (or OAPI), and after the name of each such State (or OAPI), the number of the parent title or parent application and the date of grant of the parent title or filing of the parent application;*

*(vi) if, in Box No. VI, there are **more than three earlier applications whose priority is claimed**: in such case, write "Continuation of Box No. VI" and indicate for each additional earlier application the same type of information as required in Box No. VI;*

*(vii) if, in Box No. VI, **the earlier application is an ARIPO application**: in such case, write "Continuation of Box No. VI", specify the number of the item corresponding to that earlier application and indicate at least one country party to the Paris Convention for the Protection of Industrial Property for which that earlier application was filed.*

*2. If, with regard to the **precautionary designation statement** contained in Box No. V, the applicant wishes to exclude any State(s) from the scope of that statement: in such case, write "Designation(s) excluded from precautionary designation statement" and indicate the name or two-letter code of each State so excluded.*

*3. If the applicant claims, in respect of any designated Office, the benefits of provisions of the national law concerning **non-prejudicial disclosures or exceptions to lack of novelty**: in such case, write "Statement concerning non-prejudicial disclosures or exceptions to lack of novelty" and furnish that statement below.*

Box No. VI PRIORITY CL[redacted] ☐ Further priority claims are indicated in the Supplemental Box.

Filing date of earlier application (day/month/year)	Number of earlier application	Where earlier application is: national application: country	regional application:* regional Office	international application: receiving Office
item (1) 19/11/1998	98 14687	FR		
item (2)				
item (3)				

☒ The receiving Office is requested to prepare and transmit to the International Bureau a certified copy of the earlier application(s) *(only if the earlier application was filed with the Office which for the purposes of the present international application is the receiving Office)* identified above as item(s): 1

* *Where the earlier application is an ARIPO application, it is mandatory to indicate in the Supplemental Box at least one country party to the Paris Convention for the Protection of Industrial Property for which that earlier application was filed (Rule 4.10(b)(ii)). See Supplemental Box.*

Box No. VII INTERNATIONAL SEARCHING AUTHORITY

Choice of International Searching Authority (ISA) *(if two or more International Searching Authorities are competent to carry out the international search, indicate the Authority chosen; the two-letter code may be used):*

ISA /

Request to use results of earlier search; reference to that search *(if an earlier search has been carried out by or requested from the International Searching Authority):*

Date (day/month/year)	Number	Country (or regional Office)
17/08/1999	98 14687	FR

Box No. VIII CHECK LIST; LANGUAGE OF FILING

This international application contains the following **number of sheets**:

request	:	4
description (excluding sequence listing part)	:	4
claims	:	1
abstract	:	1
drawings	:	2
sequence listing part of description	:	
Total number of sheets	:	12

This international application is **accompanied by** the item(s) marked below:

1. ☒ fee calculation sheet
2. ☐ separate signed power of attorney
3. ☐ copy of general power of attorney; reference number, if any:
4. ☐ statement explaining lack of signature
5. ☐ priority document(s) identified in Box No. VI as item(s):
6. ☐ translation of international application into (language):
7. ☐ separate indications concerning deposited microorganism or other biological material
8. ☐ nucleotide and/or amino acid sequence listing in computer readable form
9. ☐ other *(specify)*:

Figure of the drawings which should accompany the abstract: 2

Language of filing of the international application: FR

Box No. IX SIGNATURE OF APPLICANT OR AGENT

Next to each signature, indicate the name of the person signing and the capacity in which the person signs (if such capacity is not obvious from reading the request).

Michel POUPON, Agent
Epinal; 12 November 1999

For receiving Office use only

1. Date of actual receipt of the purported international application:
2. Drawings: ☐ received: ☐ not received:
3. Corrected date of actual receipt due to later but timely received papers or drawings completing the purported international application:
4. Date of timely receipt of the required corrections under PCT Article 11(2):
5. International Searching Authority (if two or more are competent): ISA /
6. ☐ Transmittal of search copy delayed until search fee is paid.

For International Bureau use only

Date of receipt of the record copy by the International Bureau:

Express Mail No. ET169352968US

PCT

REQUEST

The undersigned requests that the present international application be processed according to the Patent Cooperation Treaty.

For receiving Office use only

International Application No.

International Filing Date 09/856311

Name of receiving Office and "PCT International Application"

Applicant's or agent's file reference *(if desired) (12 characters maximum)* BRL 7 PCT

Box No. I TITLE OF INVENTION METHOD FOR PRODUCING MECHANICAL PARTS BY BREAKING DOWN INTO LAYERS WITH TURN-OVER

Box No. II APPLICANT

Name and address: *(Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country. The country of the address indicated in this Box is the applicant's State (that is, country) of residence if no State of residence is indicated below.)*

C.I.R.T.E.S. (Partnership Law 1901)
(Centre d'Ingénierie de Recherche et de Transfert de l'ESSTIN à Saint-Dié)
29 Bis rue d'Hellieule
88100 SAINT DIE

☐ This person is also inventor.

Telephone No.

Facsimile No.

Teleprinter No.

State *(that is, country)* of nationality: FRANCE

State *(that is, country)* of residence: FRANCE

This person is applicant for the purposes of: ☐ all designated States ☒ all designated States except the United States of America ☐ the United States of America only ☐ the States indicated in the Supplemental Box

Box No. III FURTHER APPLICANT(S) AND/OR (FURTHER) INVENTOR(S)

Name and address: *(Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country. The country of the address indicated in this Box is the applicant's State (that is, country) of residence if no State of residence is indicated below.)*

BARLIER, Claude
67 Chemin de la Roche
88100 COINCHES

This person is:

☐ applicant only

☒ applicant and inventor

☐ inventor only *(If this check-box is marked, do not fill in below.)*

State *(that is, country)* of nationality: FRANCE

State *(that is, country)* of residence: FRANCE

This person is applicant for the purposes of: ☐ all designated States ☐ all designated States except the United States of America ☒ the United States of America only ☐ the States indicated in the Supplemental Box

☐ Further applicants and/or (further) inventors are indicated on a continuation sheet.

Box No. IV AGENT OR COMMON REPRESENTATIVE; OR ADDRESS FOR CORRESPONDENCE

The person identified below is hereby/has been appointed to act on behalf of the applicant(s) before the competent International Authorities as: ☒ agent ☐ common representative

Name and address: *(Family name followed by given name; for a legal entity, full official designation. The address must include postal code and name of country.)*

POUPON, Michel
Cabinet Michel POUPON
3 rue Ferdinand Brunot
88026 EPINAL CEDEX
FRANCE

Telephone No. 03.29.64.05.93

Facsimile No. 03.29.64.17.33

Teleprinter No.

☐ **Address for correspondence:** Mark this check-box where no agent or common representative is/has been appointed and the space above is used instead to indicate a special address to which correspondence should be sent.

Box No.V DESIGNATION OF STATES

The following designations are hereby made under Rule 4.9(a) *(mark the applicable check-boxes; at least one must be marked)*:

Regional Patent

- ☒ **AP** **ARIPO Patent:** **GH** Ghana, **GM** Gambia, **KE** Kenya, **LS** Lesotho, **MW** Malawi, **SD** Sudan, **SL** Sierra Leone, **SZ** Swaziland, **UG** Uganda, **ZW** Zimbabwe, and any other State which is a Contracting State of the Harare Protocol and of the PCT
- ☒ **EA** **Eurasian Patent:** **AM** Armenia, **AZ** Azerbaijan, **BY** Belarus, **KG** Kyrgyzstan, **KZ** Kazakhstan, **MD** Republic of Moldova, **RU** Russian Federation, **TJ** Tajikistan, **TM** Turkmenistan, and any other State which is a Contracting State of the Eurasian Patent Convention and of the PCT
- ☒ **EP** **European Patent:** **AT** Austria, **BE** Belgium, **CH and LI** Switzerland and Liechtenstein, **CY** Cyprus, **DE** Germany, **DK** Denmark, **ES** Spain, **FI** Finland, **FR** France, **GB** United Kingdom, **GR** Greece, **IE** Ireland, **IT** Italy, **LU** Luxembourg, **MC** Monaco, **NL** Netherlands, **PT** Portugal, **SE** Sweden, and any other State which is a Contracting State of the European Patent Convention and of the PCT
- ☒ **OA** **OAPI Patent:** **BF** Burkina Faso, **BJ** Benin, **CF** Central African Republic, **CG** Congo, **CI** Côte d'Ivoire, **CM** Cameroon, **GA** Gabon, **GN** Guinea, **GW** Guinea-Bissau, **ML** Mali, **MR** Mauritania, **NE** Niger, **SN** Senegal, **TD** Chad, **TG** Togo, and any other State which is a member State of OAPI and a Contracting State of the PCT *(if other kind of protection or treatment desired, specify on dotted line)* .....

National Patent *(if other kind of protection or treatment desired, specify on dotted line)*:

- ☒ **AE** United Arab Emirates
- ☒ **AL** Albania
- ☒ **AM** Armenia
- ☒ **AT** Austria
- ☒ **AU** Australia
- ☒ **AZ** Azerbaijan
- ☒ **BA** Bosnia and Herzegovina
- ☒ **BB** Barbados
- ☒ **BG** Bulgaria
- ☒ **BR** Brazil
- ☒ **BY** Belarus
- ☒ **CA** Canada
- ☒ **CH and LI** Switzerland and Liechtenstein
- ☒ **CN** China
- ☒ **CU** Cuba
- ☒ **CZ** Czech Republic
- ☒ **DE** Germany
- ☒ **DK** Denmark
- ☒ **EE** Estonia
- ☒ **ES** Spain
- ☒ **FI** Finland
- ☒ **GB** United Kingdom
- ☒ **GD** Grenada
- ☒ **GE** Georgia
- ☒ **GH** Ghana
- ☒ **GM** Gambia
- ☒ **HR** Croatia
- ☒ **HU** Hungary
- ☒ **ID** Indonesia
- ☒ **IL** Israel
- ☒ **IN** India
- ☒ **IS** Iceland
- ☒ **JP** Japan
- ☒ **KE** Kenya
- ☒ **KG** Kyrgyzstan
- ☒ **KP** Democratic People's Republic of Korea
- ☒ **KR** Republic of Korea
- ☒ **KZ** Kazakhstan
- ☒ **LC** Saint Lucia
- ☒ **LK** Sri Lanka
- ☒ **LR** Liberia
- ☒ **LS** Lesotho
- ☒ **LT** Lithuania
- ☒ **LU** Luxembourg
- ☒ **LV** Latvia
- ☒ **MD** Republic of Moldova
- ☒ **MG** Madagascar
- ☒ **MK** The former Yugoslav Republic of Macedonia
- ☒ **MN** Mongolia
- ☒ **MW** Malawi
- ☒ **MX** Mexico
- ☒ **NO** Norway
- ☒ **NZ** New Zealand
- ☒ **PL** Poland
- ☒ **PT** Portugal
- ☒ **RO** Romania
- ☒ **RU** Russian Federation
- ☒ **SD** Sudan
- ☒ **SE** Sweden
- ☒ **SG** Singapore
- ☒ **SI** Slovenia
- ☒ **SK** Slovakia
- ☒ **SL** Sierra Leone
- ☒ **TJ** Tajikistan
- ☒ **TM** Turkmenistan
- ☒ **TR** Turkey
- ☒ **TT** Trinidad and Tobago
- ☒ **UA** Ukraine
- ☒ **UG** Uganda
- ☒ **US** United States of America
- ☒ **UZ** Uzbekistan
- ☒ **VN** Viet Nam
- ☒ **YU** Yugoslavia
- ☒ **ZA** South Africa
- ☒ **ZW** Zimbabwe

Check-boxes reserved for designating States which have become party to the PCT after issuance of this sheet:

- ☒ CR Costa Rica
- ☒ DM Dominica
- X Morocco (MA)

Precautionary Designation Statement: In addition to the designations made above, the applicant also makes under Rule 4.9(b) all other designations which would be permitted under the PCT except any designation(s) indicated in the Supplemental Box as being excluded from the scope of this statement. The applicant declares that those additional designations are subject to confirmation and that any designation which is not confirmed before the expiration of 15 months from the priority date is to be regarded as withdrawn by the applicant at the expiration of that time limit. *(Confirmation of a designation consists of the filing of a notice specifying that designation and the payment of the designation and confirmation fees. Confirmation must reach the receiving Office within the 15-month time limit.)*

Supplemental Box *If the Supplemental Box is not used, this sheet should not be included in the request.*

*1. If, in any of the Boxes, **the space is insufficient** to furnish all the information: in such case, write "Continuation of Box No. ..." [indicate the number of the Box] and furnish the information in the same manner as required according to the captions of the Box in which the space was insufficient, in particular:*

*(i) **if more than two persons are involved as applicants and/or inventors** and no "continuation sheet" is available: in such case, write "Continuation of Box No. III" and indicate for each additional person the same type of information as required in Box No. III. The country of the address indicated in this Box is the applicant's State (that is, country) of residence if no State of residence is indicated below;*

*(ii) if, in Box No. II or in any of the sub-boxes of Box No. III, the indication **"the States indicated in the Supplemental Box"** is checked: in such case, write "Continuation of Box No. II" or "Continuation of Box No. III" or "Continuation of Boxes No. II and No. III" (as the case may be), indicate the name of the applicant(s) involved and, next to (each) such name, the State(s) (and/or, where applicable, ARIPO, Eurasian, European or OAPI patent) for the purposes of which the named person is applicant;*

*(iii) if, in Box No. II or in any of the sub-boxes of Box No. III, **the inventor or the inventor/applicant is not inventor for the purposes of all designated States or for the purposes of the United States of America**: in such case, write "Continuation of Box No. II" or "Continuation of Box No. III" or "Continuation of Boxes No. II and No. III" (as the case may be), indicate the name of the inventor(s) and, next to (each) such name, the State(s) (and/or, where applicable, ARIPO, Eurasian, European or OAPI patent) for the purposes of which the named person is inventor;*

*(iv) if, in addition to the agent(s) indicated in Box No. IV, there are **further agents**: in such case, write "Continuation of Box No. IV" and indicate for each further agent the same type of information as required in Box No. IV;*

*(v) if, in Box No. V, the name of any State (or OAPI) is accompanied by the indication "**patent of addition**," or "**certificate of addition**," or if, in Box No. V, the name of the United States of America is accompanied by an indication "**continuation**" or "**continuation-in-part**": in such case, write "Continuation of Box No. V" and the name of each State involved (or OAPI), and after the name of each such State (or OAPI), the number of the parent title or parent application and the date of grant of the parent title or filing of the parent application;*

*(vi) if, in Box No. VI, there are **more than three earlier applications whose priority is claimed**: in such case, write "Continuation of Box No. VI" and indicate for each additional earlier application the same type of information as required in Box No. VI;*

*(vii) if, in Box No. VI, **the earlier application is an ARIPO application**: in such case, write "Continuation of Box No. VI", specify the number of the item corresponding to that earlier application and indicate at least one country party to the Paris Convention for the Protection of Industrial Property for which that earlier application was filed.*

*2. If, with regard to the **precautionary designation statement** contained in Box No. V, the applicant wishes to exclude any State(s) from the scope of that statement: in such case, write "Designation(s) excluded from precautionary designation statement" and indicate the name or two-letter code of each State so excluded.*

*3. If the applicant claims, in respect of any designated Office, the benefits of provisions of the national law concerning **non-prejudicial disclosures or exceptions to lack of novelty**: in such case, write "Statement concerning non-prejudicial disclosures or exceptions to lack of novelty" and furnish that statement below.*

Box No. VI PRIORITY CLAIM ☐ Further priority claims are indicated in the Supplemental Box.

Filing date of earlier application (day/month/year)	Number of earlier application	Where earlier application is:		
		national application: country	regional application:* regional Office	international application: receiving Office
item (1) 19/11/1998	98 14688	FR		
item (2)				
item (3)				

☒ The receiving Office is requested to prepare and transmit to the International Bureau a certified copy of the earlier application(s) *(only if the earlier application was filed with the Office which for the purposes of the present international application is the receiving Office)* identified above as item(s): 1

* *Where the earlier application is an ARIPO application, it is mandatory to indicate in the Supplemental Box at least one country party to the Paris Convention for the Protection of Industrial Property for which that earlier application was filed (Rule 4.10(b)(ii)). See Supplemental Box.*

Box No. VII INTERNATIONAL SEARCHING AUTHORITY

Choice of International Searching Authority (ISA) *(if two or more International Searching Authorities are competent to carry out the international search, indicate the Authority chosen; the two-letter code may be used):*

ISA /

Request to use results of earlier search; reference to that search *(if an earlier search has been carried out by or requested from the International Searching Authority):*

Date *(day/month/year)*	Number	Country *(or regional Office)*
17/08/1999	98 14688	FR

Box No. VIII CHECK LIST; LANGUAGE OF FILING

This international application contains the following **number of sheets**:

request	:	4
description (excluding sequence listing part)	:	3
claims	:	1
abstract	:	1
drawings	:	1
sequence listing part of description	:	
Total number of sheets	:	10

This international application is **accompanied by** the item(s) marked below:

1. ☒ fee calculation sheet
2. ☐ separate signed power of attorney
3. ☐ copy of general power of attorney; reference number, if any:
4. ☐ statement explaining lack of signature
5. ☐ priority document(s) identified in Box No. VI as item(s):
6. ☐ translation of international application into (language):
7. ☐ separate indications concerning deposited microorganism or other biological material
8. ☐ nucleotide and/or amino acid sequence listing in computer readable form
9. ☐ other *(specify)*:

Figure of the drawings which should accompany the abstract:

Language of filing of the international application: FR

Box No. IX SIGNATURE OF APPLICANT OR AGENT

Next to each signature, indicate the name of the person signing and the capacity in which the person signs (if such capacity is not obvious from reading the request).

Michel POUPON, Agent
Epinal; 12 November 1999

For receiving Office use only

1. Date of actual receipt of the purported international application:
2. Drawings: ☐ received: ☐ not received:
3. Corrected date of actual receipt due to later but timely received papers or drawings completing the purported international application:
4. Date of timely receipt of the required corrections under PCT Article 11(2):
5. International Searching Authority (if two or more are competent): ISA /
6. ☐ Transmittal of search copy delayed until search fee is paid.

For International Bureau use only

Date of receipt of the record copy by the International Bureau:

10811

TRAITE COOPERATION EN MATIE DE BREVETS

PCT

REC'D 12 FEB 2001

PCT

RAPPORT D'EXAMEN PRELIMINAIRE INTERNATIONAL

(article 36 et règle 70 du PCT)

15

Référence du dossier du déposant ou du mandataire BRL 8 PCT	**POUR SUITE A DONNER**	voir la notification de transmission du rapport d'examen préliminaire international (formulaire PCT/IPEA/416)
Demande internationale n° PCT/FR99/02790	Date du dépot international *(jour/mois/année)* 15/11/1999	Date de priorité *(jour/mois/année)* 19/11/1998
Classification internationale des brevets (CIB) ou à la fois classification nationale et CIB G05B19/4099		
Déposant C.I.R.T.E.S. et al		

1. Le présent rapport d'examen préliminaire international, établi par l'administaration chargée de l'examen préliminaire international, est transmis au déposant conformément à l'article 36.

2. Ce RAPPORT comprend 6 feuilles, y compris la présente feuille de couverture.

   ☐ Il est accompagné d'ANNEXES, c'est-à-dire de feuilles de la description, des revendications ou des dessins qui ont été modifiées et qui servent de base au présent rapport ou de feuilles contenant des rectifications faites auprès de l'administration chargée de l'examen préliminaire international (voir la règle 70.16 et l'instruction 607 des Instructions administratives du PCT).

   Ces annexes comprennent feuilles.

3. Le présent rapport contient des indications relatives aux points suivants:

   - I ☒ Base du rapport
   - II ☐ Priorité
   - III ☐ Absence de formulation d'opinion quant à la nouveauté, l'activité inventive et la possibilité d'application industrielle
   - IV ☐ Absence d'unité de l'invention
   - V ☒ Déclaration motivée selon l'article 35(2) quant à la nouveauté, l'activité inventive et la possibilité d'application industrielle; citations et explications à l'appui de cette déclaration
   - VI ☐ Certains documents cités
   - VII ☒ Irrégularités dans la demande internationale
   - VIII ☐ Observations relatives à la demande internationale

Date de présentation de la demande d'examen préliminaire internationale	Date d'achèvement du présent rapport
15/06/2000	08.02.2001
Nom et adresse postale de l'administration chargée de l'examen préliminaire international: Office européen des brevets D-80298 Munich Tél. +49 89 2399 - 0 Tx: 523656 epmu d Fax: +49 89 2399 - 4465	Fonctionnaire autorisé Ferla, M N° de téléphone +49 89 2399 2275

Formulaire PCT/IPEA/409 (feuille de couverture) (janvier 1994)

RAPPORT D'EXAMEN PRÉLIMINAIRE INTERNATIONAL

Demande internationale n° PCT/FR99/02790

I. Base du rapport

1. Ce rapport a été rédigé sur la base des éléments ci-après (*les feuilles de remplacement qui ont été remises à l'office récepteur en réponse à une invitation faite conformément à l'article 14 sont considérées dans le présent rapport comme "initialement déposées" et ne sont pas jointes en annexe au rapport puisqu'elles ne contiennent pas de modifications (règles 70.16 et 70.17).*) :

Description, pages:

1-5 version initiale

Revendications, N°:

1-8 version initiale

Dessins, feuilles:

1/2-2/2 version initiale

2. En ce qui concerne la **langue**, tous les éléments indiqués ci-dessus étaient à la disposition de l'administration ou lui ont été remis dans la langue dans laquelle la demande internationale a été déposée, sauf indication contraire donnée sous ce point.

   Ces éléments étaient à la disposition de l'administration ou lui ont été remis dans la langue suivante: , qui est :

   ☐ la langue d'une traduction remise aux fins de la recherche internationale (selon la règle 23.1(b)).
   ☐ la langue de publication de la demande internationale (selon la règle 48.3(b)).
   ☐ la langue de la traduction remise aux fins de l'examen préliminaire internationale (selon la règle 55.2 ou 55.3).

3. En ce qui concerne les **séquences de nucléotides ou d'acide aminés** divulguées dans la demande internationale (le cas échéant), l'examen préliminaire internationale a été effectué sur la base du listage des séquences :

   ☐ contenu dans la demande internationale, sous forme écrite.
   ☐ déposé avec la demande internationale, sous forme déchiffrable par ordinateur.
   ☐ remis ultérieurement à l'administration, sous forme écrite.
   ☐ remis ultérieurement à l'administration, sous forme déchiffrable par ordinateur.
   ☐ La déclaration, selon laquelle le listage des séquences par écrit et fourni ultérieurement ne va pas au-delà de la divulgation faite dans la demande telle que déposée, a été fournie.
   ☐ La déclaration, selon laquelle les informations enregistrées sous déchiffrable par ordinateur sont identiques à celles du listages des séquences Présenté par écrit, a été fournie.

4. Les modifications ont entraîné l'annulation :

- ☐ de la description, pages :
- ☐ des revendications, nos :
- ☐ des dessins, feuilles :

5. ☐ Le présent rapport a été formulé abstraction faite (de certaines) des modifications, qui ont été considérées comme allant au-delà de l'exposé de l'invention tel qu'il a été déposé, comme il est indiqué ci-après (règle 70.2(c)) :

   (Toute feuille de remplacement comportant des modifications de cette nature doit être indiquée au point 1 et annexée au présent rapport)

6. Observations complémentaires, le cas échéant :

V. Déclaration motivée selon l'article 35(2) quant à la nouveauté, l'activité inventive et la possibilité d'application industrielle; citations et explications à l'appui de cette déclaration

1. Déclaration

Nouveauté	Oui :	Revendications	
	Non :	Revendications	1
Activité inventive	Oui :	Revendications	
	Non :	Revendications	2-6
Possibilité d'application industrielle	Oui :	Revendications	1-6
	Non :	Revendications	

2. Citations et explications
   voir feuille séparée

VII. Irrégularités dans la demande internationale

Les irrégularités suivantes, concernant la forme ou le contenu de la demande internationale, ont été constatées :
voir feuille séparée

Les documents suivants figureront dans la présente notification avec le numéro d'ordre attribués ci-après:

D1: US 5 354 414 (A) (non cité dans le rapport de recherche)
D2: US 4 001 069 (A)
D3: WO 95/08416 (A)

Point V

1. La présente demande ne remplit pas les conditions énoncées à l'***article 33(2) PCT***, l'objet des *revendications indépendantes **1, 7 et 8*** n'étant pas nouveau pour les raisons suivantes:

   1.1 Le document **D1**, qui est considéré comme l'état de la technique le plus proche, décrit un procédé de prototypage rapide ayant toutes les caractéristiques du préambule de la *revendication **1***, comme apparent dans l'abrégé de D1. En outre, les caractéristiques spécifiées dans la partie caractérisante de la revendication 1, relatives à la structure des strates formées par une partie centrale, une partie extérieure et une interface de liaison entre les deux, sont mises en évidence dans plusieurs passages de D1 (cf, *colonne 18, lignes 60-62; colonne 19, lignes 23-37*).

   1.2 Les mêmes objections s'appliquent également aux *revendications **7 et 8***, le document **D1** décrivant les pièces obtenues par le procédé décrit ci-dessus (cf, *abrégé; figures 36, 40; colonne 18, ligne 54 - colonne 19, ligne 9*).

2. Les *revendications dépendantes **2 - 6*** ne semblent pas contenir de caractéristique supplémentaire qui, en combinaison avec l'objet de l'une quelconque des revendications dont elles dépendent, impliquerait une activité inventive. Les raisons en sont les suivantes:

2.1 La caractéristique indiquée dans les *revendications 2 et 3* concernant la provision d'orifices dans les strates pour faciliter leur positionnement correcte lors de l'assemblage, est décrite dans **D1** (cf, *colonne 18, ligne 62 - colonne 19, ligne 9*). Il est à noter que la sélection d'une forme particulière des orifices, soit circulaire soit à section géométrique polygonale, telle qu'indiqué dans les revendications 2 et 3, est considérée comme un choix que la personne de métier effectuerait sans qu'une activité inventive soit impliquée.

2.2 La pièce obtenue en utilisant la technique décrite dans **D1** (cf, *colonne 18, ligne 62 - colonne 19, ligne 9*) présente une structure autoporteuse telle qu'indiquée dans la *revendication 4*.

2.3 Les différents schémas de montage avec les outils correspondants (plaque de montage, tige insert) définis dans les *revendications 5 et 6* constituent des solutions bien connues utilisées pour l'assemblage des strates dans le domaine du prototypage rapide (cf, **D2**, *figures 5-10; colonne 4, ligne 49 - colonne 5, ligne 17; D3, page 15, lignes 3-20*).
Par conséquent, le choix des types d'assemblage indiqués dans les revendications mentionnées ci-dessus est seulement une des possibilités que la personne du métier pourrait choisir, selon le cas d'espèce, parmi plusieurs possibilités évidentes, sans qu'une activité inventive soit impliquée.

Point VII

1. Ce rapport termine la procédure internationale. Par conséquent, les observations suivantes visent à assister le Demandeur lors d'une éventuelle décision d'entrer dans la phase régionale, en particulier devant l'OEB.

1.1 En vue de satisfaire aux conditions énoncées à la ***règle 6.3(b) PCT***, la revendication indépendante devrait être présentée en deux parties, les caractéristiques qui sont comprises dans l'état de la technique (cf. document D1) étant indiquées dans la première partie, et les caractéristiques supplémentaires pour lesquelles la protection est recherchée étant indiquées

dans la partie caractérisante.
Les revendications dépendantes doivent, le cas échéant, être modifiées en accord avec la revendication 1 à déposer.

1.2 En vue de faciliter la compréhension des revendications, des signes de référence devraient être mis entre parenthèses dans toutes les revendications (***règle 6.2(b) PCT***), et ceci dans les deux parties des revendications.

1.3 En vue de satisfaire aux conditions énoncées à la ***règle 5.1(a)(ii) PCT***, il appartient au demandeur d'harmoniser le texte de la description avec celui des nouvelles revendications à déposer, de citer dans la description les documents D1, D2 et D3 et d'indiquer l'état correspondant de la technique.

*****

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 19/00 — A1

(11) International Publication Number: WO 97/07474

(43) International Publication Date: 27 February 1997 (27.02.97)

(21) International Application Number: PCT/US96/13486

(22) International Filing Date: 21 August 1996 (21.08.96)

(30) Priority Data:
517,092 21 August 1995 (21.08.95) US

(71) Applicant: UNIVERSITY OF UTAH RESEARCH FOUNDATION [US/US]; 210 Park Building, Salt Lake City, UT 84112 (US).

(72) Inventors: THOMAS, Charles, L.; 217 North 900 West, Salt Lake City, UT 84116 (US). WANG, Zetian; 1901 East Sunnyside Avenue #638, Salt Lake City, UT 84108 (US).

(74) Agents: BOND, Laurence, B. et al.; Trask, Britt & Rossa, P.O. Box 2550, Salt Lake City, UT 84110 (US).

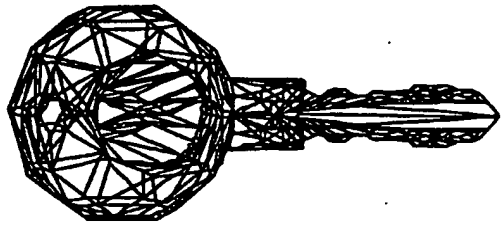
(81) Designated States: CA, JP, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

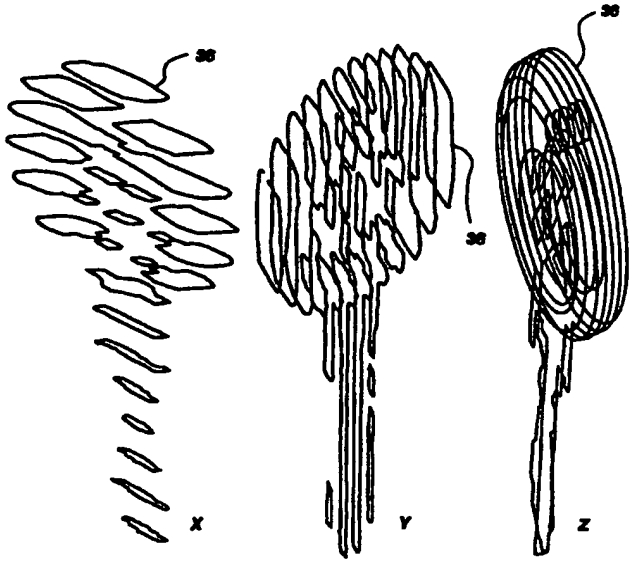
Published
With international search report.

(54) Title: IMPROVED RAPID PROTOTYPING METHOD

(57) Abstract

A rapid prototype modeling system operates to first electronically decompose a discrete part represented by a stereolithography file into thick layers (36), which are then electronically sliced into cross-sectional slices (39) the thickness of a sheet (38) of construction material. The slices (39) are cut from sheets (38) of the construction material in a pattern which permits construction of the layers (36) by stacking the sheets (38). The layers (36) are then stacked appropriately to create a physical model of the discrete part.





FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	**GB**	United Kingdom	**MW**	Malawi
AT	Austria	**GE**	Georgia	**MX**	Mexico
AU	Australia	**GN**	Guinea	**NE**	Niger
BB	Barbados	**GR**	Greece	**NL**	Netherlands
BE	Belgium	**HU**	Hungary	**NO**	Norway
BF	Burkina Faso	**IE**	Ireland	**NZ**	New Zealand
BG	Bulgaria	**IT**	Italy	**PL**	Poland
BJ	Benin	**JP**	Japan	**PT**	Portugal
BR	Brazil	**KE**	Kenya	**RO**	Romania
BY	Belarus	**KG**	Kyrgystan	**RU**	Russian Federation
CA	Canada	**KP**	Democratic People's Republic of Korea	**SD**	Sudan
CF	Central African Republic			**SE**	Sweden
CG	Congo	**KR**	Republic of Korea	**SG**	Singapore
CH	Switzerland	**KZ**	Kazakhstan	**SI**	Slovenia
CI	Côte d'Ivoire	**LI**	Liechtenstein	**SK**	Slovakia
CM	Cameroon	**LK**	Sri Lanka	**SN**	Senegal
CN	China	**LR**	Liberia	**SZ**	Swaziland
CS	Czechoslovakia	**LT**	Lithuania	**TD**	Chad
CZ	Czech Republic	**LU**	Luxembourg	**TG**	Togo
DE	Germany	**LV**	Latvia	**TJ**	Tajikistan
DK	Denmark	**MC**	Monaco	**TT**	Trinidad and Tobago
EE	Estonia	**MD**	Republic of Moldova	**UA**	Ukraine
ES	Spain	**MG**	Madagascar	**UG**	Uganda
FI	Finland	**ML**	Mali	**US**	United States of America
FR	France	**MN**	Mongolia	**UZ**	Uzbekistan
GA	Gabon	**MR**	Mauritania	**VN**	Viet Nam

IMPROVED RAPID PROTOTYPING METHOD

This application includes a computer program listing printout submitted in the form of an appendix.

Technical Field: This invention relates to prototype modeling techniques. It is particularly directed to the construction of solid patterns derived from 3D CAD software-generated models.

Background Art: Current rapid prototyping (RP) technology offers many advantages over more traditional prototyping techniques. For example, RP decreases the time required to produce an initial prototype. Currently available RP systems are generally flexible, and produce prototypes of superior accuracy. Unfortunately, they are also significantly more expensive to utilize than are traditional systems. Available techniques include stereolithography, laminated object manufacturing, fused deposition modeling, selective laser sintering, and ballistic particle manufacturing. These techniques in general operate on a common paradigm. They each decompose the part into a series of layers or cross sections. The part is then constructed by sequentially creating each layer and bonding it to the previous layer. This bonding usually occurs automatically as a part of the layer creation process. In the case of laminated object manufacturing, each layer is bonded prior to cutting the outline of the cross section. In any event, while several different parts may be built simultaneously on a common build platform, each layer of each discrete part must be created in sequential order.

Traditional RP techniques incorporate various techniques for dealing with cantilever overhangs in the part during construction. A support structure is often created in stereolithography systems. In some systems, the excess material surrounding each cross section provides the necessary support. Parts with complex curves have little contact area with the support structure during the construction procedures of many existing RP techniques. For example, a sphere theoretically contacts the build platform at a single point. As the sphere is built up from this point contact, it can easily shift, destroying the registration of the layers. For a more generalized part, the operator must decide how to orient the part, attempting to minimize overhangs and maximize the support provided by the building platform.

Construction of prototype parts by the application of RP technology usually requires the use of high-priced materials. Prohibitive costs have limited the proliferation of RP application to well-financed users, typically large companies with correspondingly substantial development budgets and RP service bureaus which demand large fees. Schools and typical smaller enterprises often cannot justify the acquisition of RP systems. There is a need for an inexpensive system capable of making RP technology directly accessible to small businesses and students.

DISCLOSURE OF INVENTION

This invention provides an inexpensive rapid prototyping method which has particular application in the field of education as well as in various industrial fields. Practice of the method is facilitated with a novel system which typically interconnects commercially available hardware and software elements through customized hardware and software elements. The cost of the complete, fully operational system is significantly less than (typically, a small fraction of) the cost of current state-of-the-art RP systems. The accuracy of a prototype fabricated using a rudimentary system of this invention is adequate to demonstrate the form of a modeled object. More refined versions may in some instances produce prototypes adequate to meet fit or function requirements. In any case, the system of this invention is affordable for schools and small businesses. It allows the designer to make a physical inspection of a design at a very low cost, typically two or three orders of magnitude below the cost associated with prototyping by currently available RP technologies.

The system of this invention creates a part from a series of cross sections as do conventional RP systems. A fundamental departure from prior systems resides in the layup iterations followed in bonding sequential layers together. This invention first creates all of the layers required, and then bonds the layers together, through a separate efficient procedure. This step, and the nature of the construction materials utilized, introduce significant additional flexibility to the final construction process. The invention avoids, for example, the limitation of existing RP systems whereby the layers must be created sequentially, one layer at a time.

According to this invention, the part is first decomposed into thick slices, or layers. These layers can be positioned to eliminate unwanted overhangs prior to cutting the outline of the cross section. The system provides added flexibility by permitting the operator to mirror selected thick layers, building any such layer either from the bottom up or the top down. Thus, when making a sphere, the first thick section will be constructed from the top down, thereby providing a large area of contact between the build platform and the part. Subsequent layers are built from bottom to top because they have adequate contact area as so oriented. This mirroring of one of the layers is then corrected for during the subsequent assembly of the thick layers.

The system operates to arrange thick layers of a single discrete part across a build platform. These layers may then be sliced to the thickness of the construction material. As a consequence, this invention can often create a part that requires many slices from few sheets of construction material. Assuming that a part is sufficiently small that a single sheet of construction material can accommodate 8 thick layers, 64 slices can be registered and bonded in 14 steps instead of 63. (The 8 sheets are registered and bonded, requiring 7 steps. Then the resulting 8 sections are registered and bonded, requiring 7 more steps.)

A typical rudimentary system of this invention comprises a plotter, a personal computer, a simple registration stand (build table) and the software (ZWSLICE) disclosed in the microfiche appendix. ZWSLICE reads three dimensional ("3-D") solid models from commercially available drawing software, and electronically cuts them into paper-thin slices. The software-generated slices are converted to physical slices cut out on the plotter. The physical slices (typically of paper, plastic or other sheet material) are then stacked on the build table to construct a physical embodiment of the computer model. These embodiments may be treated as prototype parts. Alternatively, they may be used as patterns for constructing molds or shells from which to cast parts from metal or other rigid material. In other instances, they may be used as paper molds from which to produce plastic parts.

The invention may be embodied as an inexpensive rapid prototyping device that creates physical models from 3D electronic computer-aided design ("CAD") models. It performs this function by generating a series of cross sections, slicing the part into many layers. Each cross section is cut from sheet construction material (usually paper) using a commercial sign making plotter. The cross sections are registered and laminated together forming the physical model. A primary advantage of the system of this invention is that the initial cost of the requisite hardware and software, as well as the cost of producing a typical part are 10 times to 20 times less expensive than existing rapid prototyping techniques. Assuming that the user already owns a computer and CAD software, the additional hardware and software required to complete the system may have a retail price on the order of a typical personal computer assembly.

Individual sheets of construction material may include a construction layer and a backing layer fixed to the construction layer with adhesive material. Individual physical slices may then be cut from the construction layer, leaving the backing layer in tact. Individual pluralities of the physical slices may then be distributed in corresponding patterns among an ordered set of construction sheets. The plotting step may include locating index positions on the sheets of construction material, and the cutting step may include the placement of registration holes at the index positions through the construction sheets. The holes facilitate the precise registration of respective pluralities of the physical slices carried by individual construction sheets within the set when all of the construction sheets are stacked in the order of the set with registration pins inserted through the registration holes. The registration holes may also be located to facilitate the precise registration of constructed layers when they are stacked to recompose the object.

Building speed is dependent upon a number of factors, including part dimensions, layer thickness, and operator skill. Typical building speeds range from about 0.2 to about 3.1 hours per vertical centimeter (one-half to about 8 hours per vertical inch). Because models can be constructed of readily available paper materials, construction costs are trivial compared to other RP systems.

BRIEF DESCRIPTION OF DRAWINGS

In the drawings, which illustrate what is currently regarded as the best mode for carrying out the invention:

FIG. 1 is a flow diagram of a typical embodiment of the invention;

FIG. 2 is a pictorial representation in two parts; FIG. 2a illustrating an STL representation of a solid model, and FIG. 2b illustrating that model sliced in the x, the y, and the z directions;

FIG. 3 is a sketch of a plot file illustrating the layout of slices of the part of FIG. 2 and registration holes on one sheet of construction material;

FIG. 4 is a pictorial view of a registration table of the invention;

FIG. 5 is a view in elevation of the registration table of FIG. 4 illustrating a stacked arrangement of the slices from a plurality of sheets of the type illustrated by FIG. 3;

FIG. 6 is a two-part view in cross section; FIG. 6a illustrating unsupported overhangs which may occur, and FIG. 6b illustrating a solution to this problem through hierarchical model decomposition;

FIG. 7 is a pictorial view of typical prototype parts which may be constructed by the system of the invention;

FIG. 8 is a two-part drawing of which 8a illustrates the layup of a pattern useful for sand casting; and FIG. 8b illustrates the cope and the drag elements of a mold;

FIG. 9 is a flow diagram illustrating an investment casting procedure utilizing prototypes constructed in accordance with the invention;

FIG. 10 is a schematic illustration of a lost foam casting utilization of a foam model produced in accordance with the invention;

FIG. 11 is a two-part drawing of which FIG. 11a illustrates an actual part, while FIG 11b illustrates a layered paper mold designed for that part;

FIG. 12 includes FIG. 12a and FIG. 12b, which are plan and elevation views, respectively, of a registration platform of the invention and FIG 12c, which is a plan view of a typical sheet of construction paper;

FIG. 13 includes FIG. 13a and FIG. 13b, which are similar to FIGS. 12a and 12b, and illustrates the manner in which a sheet such as that illustrated by FIG. 12c may be positioned on a registration table of the invention;

FIG. 14 includes FIGS. 14a and 14b, which are similar to FIGS. 13a and 13b, respectively, and illustrate a selected registration hole placement;

FIG. 15 includes FIGS. 15a and 15b, which are similar to FIGS. 14a and 14b, respectively, and illustrate an alternative registration hole placement;

FIG. 16 includes FIGS. 16a and 16b, which are similar to FIGS. 14a and 14b, respectively, and illustrate a partially constructed model; and

FIG. 17 includes FIGS. 17a and 17b, which are similar to FIGS. 16a and 16b, respectively, and illustrate another partially constructed model.

BEST MODE FOR CARRYING OUT THE INVENTION

FIG. 1 outlines the operation of a typical embodiment of the invention. As shown, a commercial CAD system, designated generally 30, is configured to provide an input file in conventional STL file format. Software, such as that disclosed in the microfiche appendix, designated generally 32, operates on the Stereolithography ("STL") file to create a Hewlett Packard Graphics Language ("HPGL") plot file and to print out all of the pages. The pages (sheets) are registered in proper sequence, and are bonded and coated to create a physical object (model or pattern) in a post process, designated generally 34.

Currently available CAD software packages capable of generating a solid model provide an output file which can be converted into an STL file format (a faceted representation of the model). The surface triangles of the STL format combine to approximate the surfaces of the model, as shown by FIG. 2a. The accuracy of a faceted representation depends upon the chord height selected for the CAD software. The chord height is the distance between an actual curve and the straight line approximation of this curve. The smaller the chord height, the more accurate the representation of the model. The minimum attainable chord height varies for each CAD software. There are many commercially available solid modeling CAD programs, such as AUTOCAD, CADKEY, and Pro/ENGINEER,

that can create the STL file format. FIG. 2a shows an example of an STL representation of a solid model of an ignition key part 33 (FIG. 7).

The software 32 (microfiche appendix) can read and display an STL file generated by any CAD program. The operator can change the orientation of a part by rotating it. The software 32 gives the maximum dimensions of the input part in the x, y, and z directions so that the operator can choose the best direction for slicing the part. After the software has finished slicing the whole object, each cross section (thick layer) is displayed on the screen so that the operator can verify that the axis of slicing minimizes the number of layers 36 and maximizes accuracy. FIG. 2b shows the part illustrated by FIG. 2a sliced in the x, the y, and the z directions. The slicing orientation resulting in the lowest number of layers 36 is usually preferred.

Referring to FIG. 3, the software 32 (FIG. 1) automatically calculates and lays out the maximum number of layers 36 that can be represented on one construction sheet 38. The thick layers 36 are sliced into parallel thin slices 39 (the thickness of the construction sheet material) arranged on the sheet 38 beginning from the bottom to the top and then from the left to the right, until the entire sheet 38 is filled.

Each sheet 38 of cross section slices 39 is sorted and converted to HPGL plot file format (32, FIG. 1), which is output to a sign making plotter (40, FIG. 1). The plotter cuts slices 39 of the part, and automatically adds registration holes 42 (FIG. 3) on each sheet. Each sheet 38 illustrated by FIG. 3 consists of two layers--a paper 43 held by adhesive 44 to a backing layer 46. Preferably, only the paper 43 is cut to produce a slice 39, leaving the backing 46 in tact. However, registration holes 42 are cut completely through the sheet 38. The operator has the option of choosing from several construction materials. The most common are readily available label paper and foam sheet materials.

Once all the sheets 38 have been cut, they are mounted in order on a registration table, generally 50, as shown in FIG. 4. The sheets 38 are registered by means of the pins 52 inserted through appropriate registration holes 42 (FIG. 3). Stacking of the sheets 38 organizes the slices 39 carried by respective stacked sheets

into reassembled layers, generally 36 (FIG 5). The sheets are usually stacked with the backing layer 46 up. As each sheet 38 is stacked, the backing layer 46 is peeled off (and excess construction layer 43 removed), leaving residual adhesive 44 on the upper surface of the registered slices 39. The slices carried by the next sheet 38 that is stacked are thus glued firmly to the previously stacked slices 39 in proper arrangement. As illustrated, the whole decomposed object is first recomposed into its component thick layers 36 (FIG. 5). These thick layers 36 may be separated and stacked, being positioned by a second set of registration holes 58 (FIG. 3) created by the software 32 to complete the construction of the solid object.

After construction, the modeled (recomposed) parts are given a coating to increase their rigidity, to prevent layer delamination, and to cover areas where the adhesive 44 is exposed. This coating may comprise paint, acrylic coating or glue, and is typically applied by spraying techniques or by dipping the recomposed part in a coating material.

A notable characteristic of the present invention is the opportunity it provides to evaluate the geometry of the part (See FIG. 7) being built and to decompose the part into a convenient number of thick layers 36, each capable of further slicing into a plurality of thin slices 39. By efficient layout of layers 36, a maximum number of slices 39 may be laid out on one sheet 38, as shown in FIG. 3. As illustrated, the part being built is small enough for eight layers 36 of slices 39, having the maximum dimensions of the part, to fit on a single sheet 38. This "hierarchical model decomposition" approach allows eight layers to be stacked simultaneously, greatly reducing build time. Another advantage of hierarchical model decomposition is its inherently efficient utilization of construction material. For example, if the part in FIG. 3 requires 70 slices 39 to build, but eight slices 39 fit on a single sheet 38, only nine sheets 38 are required, rather than the conventional 70 sheets. The nine sheets 38 may be stacked first, resulting in eight layers 36. The eight layers 36 may then be stacked in order. That procedure requires a total of 15 stacking iterations, rather than the 70 iterations which would otherwise be required.

As the part is stacked, there are potentially unsupported overhangs, generally 60, as shown in FIG. 6a. If the overhang is large, and many layers are stacked on top of it, the inherent cantilever effect tends to bend the part at the overhang, as indicated in phantom 62. This problem is solved with hierarchical model decomposition, whereby a parting line 64 can be defined by the operator right at the overhang as illustrated in FIG. 6b.

Often layers built from the bottom up, such as shown by FIG. 6a, develop overhangs 60. It is often preferable for such parts to be built upside down, as shown by FIG. 6b. Sometimes, it is preferable for only certain sections of a part to be built upside down, and others to be built right side up. These arrangements can be accomplished if the part, or a section of the part, is mirrored, so that the top faces down, and the parts are stacked with the top going down first, and the bottom going down last, as shown in FIG. 6b. This construction approach eliminates overhang and improves the stability of the part during construction. It is generally preferable for layers 36 to be organized such that smaller slices 39 are stacked on top of larger slices 39 during the layup procedure.

EXAMPLES

Several prototype parts were built following the system illustrated by FIG. 1 and utilizing the software disclosed in the microfiche appendix. The parts discussed in the following examples are shown in FIG. 7, and were built out of 0.0127 centimeters (0.005 in.) thick label paper.

I. Screw driver handle

The screw driver handle 70 shown in FIG. 7 was drawn in Silverscreen, a 3-D modeling software. This example demonstrates the advantages of hierarchical model decomposition and mirroring, both of which are provided by the software 32. The part 70 was made from paper construction material, utilizing 207 slices. Registering and bonding these slices sequentially would have been very tedious. The part was decomposed into 8 thick layers which were built simultaneously. The

207 slices were thus cut from 26 sheets, reducing the number of registration and bonding operations from 207 to 32.

As can be seen from FIG. 7, any outside slice of the part 70 will inherently have very little surface area. To avoid creating overhangs (See FIG 6a), the software 32 mirrored the initial outer section. The thicker slices could then be layed up first. The mirroring action required the software 32 to reverse the location of the registration holes 58 created by the software. Hence, the mirroring was automatically corrected when the thick layers 36 were subsequently assembled. The total build time for this part was about 3 hours.

II. Ignition key

The ignition key 33, also shown in FIG. 7, was drawn in Pro/ENGINEER. This example demonstrates the advantage of being able to define parting line(s) while decomposing the model. It also utilized hierarchical model decomposition. The need for building support blocks was eliminated by defining a parting line through the center of the overhanging portion of the key. This parting line avoided overhanging portions. The whole model was decomposed into 61 slices 39 organized into six thick layers 36. The total build time for this part was about 2 hours.

III. Calculator

This part (72, FIG. 7) was drawn in Pro/ENGINEER. The part was decomposed into 61 slices organized into 6 thick layers which were built simultaneously. Construction of this part presented a particular problem when registering the keyboard keys 73. The slices containing the key cross sections were stacked on top of the base slices. This arrangement caused the base which had its sticky side up and extra material around the keys to stick together. This problem was solved by either covering the exposed area on the base layer with a non stick surface or by peeling the unwanted material from each of the layers containing the key cross sections before bonding. The total build time for this part was about 3 hours.

It is often desirable to create a prototype that has significant physical properties not provided by the construction materials used to fashion patterns in accordance with FIG. 1. Those patterns can be converted to metal parts, either directly or following enhancements of the kind routinely followed by pattern makers more closely to match the specifications of a desired finished part. Conventional techniques exist to create metal parts from prototypes made of foam and paper. Investment casting, sand casting, and lost foam casting techniques may be followed to produce metal parts from the patterns produced by this invention.

IV. Sand casting

Sand casting can be used to obtain metal parts from paper models constructed in accordance with this invention. FIGs 8a and 8b illustrate the manner in which the desired model is built from the CAD drawing 30 in two halves with each half 74 separated by a parting line 76 which is defined by the designer. As illustrated by FIG. 8a, two holes 80 are created on each of these pieces for registration. These parts are then coated with a suitable material to prevent contact between the adhesive surface of the part and the sand. Each half is then registered and bonded to the base 82 of the cope and the drag using two removable pins (not shown) provided on the base of each half through the holes 80. The standard sand casting procedure can then be followed by ramming sand in the cope and the drag with the runners and the pattern in place. After the ramming, the pattern can be removed from each of the two halves resulting in the required sand cavity. The cope and the drag can then be assembled and casts can be made.

V. Investment casting

Investment casting is usually used to create metal parts from prototypes made out of foam or wax. This process can be implemented on foam models constructed by the software 32 of this invention. As shown by FIG. 9, the desired part is prototyped from a CAD model using layers of foam. The prototype is then directly dipped into a slurry of refractory material until a thick ceramic coating is obtained. This mold is then heated, first in an inverted position at a temperature of

93°C to 191° (200°F to 375°F) for about 12 hours, and then in the upright position at a temperature of 649°C to 1038°C (1200°F to 1900°F) to completely melt away the foam prototype and create a hollow ceramic shell. Molten metal can then be poured into the hollow portion to obtain the finished casting.

VI. Lost foam (Evaporative pattern) casting

Lost foam casting can be applied to obtain metal parts from models created by this invention as illustrated by FIG. 10. The desired prototype is first drawn in a CAD software. A sprue and a vent are then added to the model within the CAD software. The software 32 can then be utilized by an operator to build a foam model 84 from this drawing. This foam part is then placed in a flask 85 and covered with sand 86. Molten metal poured into the sprue 87 instantaneously evaporates the foam resulting in the metal filling the volume previously occupied by the foam.

VII. Paper molds for plastic parts

Prototype paper molds for parts drawn on the computer can be used to make parts from any room temperature molding material as shown by FIG. 11. Starting with a part 90, a mold 92 may be designed using Pro/Mold design, a module of Pro/ENGINEER. The mold 92 may be made up of two halves with a parting surface 94 that is defined by the designer. Two registration holes 96 are also created on each of these halves. The two pieces are built using procedures illustrated by FIG. 1 and the software of the microfiche appendix. They are then coated with a suitable material to prevent adhesion between the paper and the molding material. The two halves are then mounted on respective base pieces 98 for increased rigidity using the two removable pins provided on each of these base pieces and the holes created in the CAD system for accurate registration. The two pieces are then registered and put together, and can be used to create parts by room temperature molding.

FIGS. 12-17 illustrate the operation of a system of this invention which consists of commercially available and custom elements arranged to provide a variety f functions. The elements of the system are as follows:

3D CAD Software:

The system is compatible with any CAD software that is capable of generating a 3D model and converting it to standard Stereolithography format. Common packages that have this capability are: CADKEY, Autocad, Pro-Engineer, and SilverScreen.

Computer to operate software:

The software (ZWSLICE) described in detail in the microfiche appendix is currently written in the DOS environment, requiring a PC compatible computer.

Sign making plotter:

A suitable such device is the PNC-9000 made by Roland Digital. Other HPGL plotters with cutting capabilities are also considered to be suitable.

Construction material:

The system works with adhesive backed sheets of material up to 1 mm in thickness. The material must be compatible with the cutting blade supplied by the plotter. For the Roland plotter, suitable materials are standard label paper available in 21.6 centimeters by 27.94 centimeters (8 1/2" by 11") by 0.0127 centimeters (0.005") thick sheets and adhesive backed polystyrene foam sheets available in 0.069 centimeters (0.027") thickness.

Adhesive spray:

Preferred building technique requires that the first layer of the part be sticky on both sides. This requirement is satisfied by coating the top of the first layer with a commercial spray adhesive.

ZWSLICE software:

This software is used to generate cross sections from the model, create plot files, and organize the cutting and assembly process. ZWSLICE provides the following capabilities:

Load and view Stereolithography format files (3D models).

Select layer orientation and preview part sliced in this orientation.

Automatically decompose part into thick layers, and arrange to cover the full sheet of construction material.

Allow operator to choose whether each thick layer will be built from bottom up or top down. (Mirror function)

Slice this array of thick layers into thin slices which are the same thickness as the construction material.

Generate HPGL plot files for each sheet of construction material including part cross sections (slices) and any necessary registration holes.

Display each cross section individually.

Display the complete plot file for each construction sheet.

Interact with operator to plot any or all of the sheets.

Store plot files for future use.

Registration system for alignment of layers:

The registration system, generally 100, shown in FIG. 12 allows a variety of registration options:

Corner Box alignment. As shown by FIG. 13, if the construction sheets 38 are carefully aligned in the plotter, the cross sections will be accurately registered to the edges of the construction sheet. Sequential sheets are aligned by aligning two edges of each sheet with the guide rails 101, 102 of the corner box, generally 104.

Registration holes outside part. FIG. 14 illustrates an arrangement in which one or more holes 42 placed outside the part 106 can be used with guide pins 52 to register sequential sheets 38.

Registration holes in part. FIG. 15 illustrates the manner in which cross section slice 39 alignment can be assured for parts having appropriate geometry by generating two registration holes 108 passing completely through each part. Sequential layers are guided by registration pins 110 extending from a pin plate 112 mounted in a recess in the platform 114, as shown.

Surface coating material:

After construction, the parts are given a coating to increase the rigidity of the parts, prevent layer delamination, and cover areas where the adhesive is exposed. Carpenter's glue thinned with tap water is useful for this purpose.

To utilize this device, the operator begins with a 3D model generated with a 3D CAD software. The file is stored in Stereolithography format. Using ZWSLICE, the file is loaded and viewed. The operator is prompted to select a slice orientation, and subsequently the part is displayed, sliced in this orientation, to verify the operator's selection. Changes are allowed at any time. When the operator is satisfied, the program evaluates the build area required for the part and then determines how many of these areas can be fit onto a single sheet (or page) of construction material.

The part is then automatically sliced into thick sections that occupy the maximum practical portion of surface area of the construction material. These thick sections are subsequently sliced into layers matching the construction material thickness, and a plot file is generated for each sheet of material required. Registration holes are automatically added to the plot file for sheet to sheet registration and for section to section (layer 36) registration. Each part cross

section (slice 39) generated can be displayed to verify the intended result. The plot file (See FIG. 3) for each sheet of construction material can also be displayed.

When the operator is satisfied with the displayed information, the construction sheets can be cut. ZWSLICE prompts the operator in the operation of the plotter to generate all or any one of the construction sheets. Ideally, the plotter cutting pen is adjusted so that a single pass of the cutter cuts through the construction material but not the backing material. Registration holes 42, 58, 108 are cut completely through the backing material; e.g. by effecting multiple passes of the cutter.

The part is assembled on the build table by sequentially registering and bonding the layers. Each sheet 38 contains two registration holes 42 corresponding to the two permanent registration pins 42 on the build table 100. The first layer and the last layer may also contain registration holes 58, 108 for each section 36 of parts that have been decomposed into sections. The build process proceeds as shown by FIG. 16.

The top of the first sheet is coated with spray adhesive, and the excess construction material is peeled from the backing sheet leaving only the part cross sections 120. This sheet is placed face up on the registration table 100 by carefully passing the registration pins 52 through the registration holes 42.

The second layer 122 is placed on the registration table face down by passing the registration pins 52 through the registration holes 42. The second sheet 122 is pressed firmly against the first sheet 120 to insure that the adhesive bonds the two layers together. The backing layer of the second sheet 122 is carefully removed, leaving the second layer cross section on the registration table, bonded to the first layer cross section. The second layer 122 is now positioned adhesive side up ready to bond to the next layer 124. (It is sometimes advisable to remove the excess construction material from the second sheet before registering and bonding this sheet. This prevents adhesive from the first layer which may overlap the excess material on the second sheet from bonding to that material.)

Subsequent layers are applied exactly as the second layer was applied. Care must be taken to insure that when the backing is peeled away the cross section

remains bonded to the registration table and the part is not peeled off the build table.

Parts that cover only a portion of the construction sheet can be sliced into thick sections 36 and assembled in a two step process. The first step proceeds as described in connection with FIG. 16, resulting in a construction sheet with several sections 125-128 adhered to its surface, as shown in FIG. 17. The second step of the assembly process proceeds as follows: When the last layer is applied, the backing material is not immediately removed. Because the part is being built in sections, it is important to note the orientation and order of the layers. ZWSLICE will display a building page that numbers the sections in order of lamination. These numbers may be marked on the upper surface backing material before the sheet holding the sections is removed from the registration table.

The entire assembly, containing an upper sheet, part sections, and a lower sheet is removed from the registration table 100. The sections 125-128 are cut apart, retaining the proper registration holes with each section.

Next, two registration pins 130 are threaded into the metal plate 112 in the center of the registration table. The spacing between the pins 130 should match that between the registration holes 132 on the sections 125-128.

The portion of the backing sheet that does not contain registration holes is peeled away from section 125. This section is placed on the registration table by passing the registration pins 52 through the registration holes 132. This section is positioned backing side down.

The backing sheet portion that does not contain registration holes is peeled away from section 126. This section is positioned on the registration table 100 by passing the registration pins 52 through the registration holes 132. This section is positioned backing side up. Section 126 should now be bonded to section 125. Sections 127, 128 and any subsequent sections are assembled just as section 126 was assembled.

Reference herein to details of the illustrated embodiments is not intended to limit the scope of the appended claims, which themselves recite those limitations regarded as definitive of the invention.

APPENDIX

```
/*****************************************************************************
  ZWSLICE1.CPP
  Shapemaker 1 software.      Version 1.0
  Copyright (c) 1995, University of Utah. All rights reserv d.
  This is the main program which directs all the functions and
  subfunctions.
  This software is compiled and linked under the Borlandc C++ 3.1
  enviroment.
  Writen by Zetian Wang

Function Description
1)  title(void);                    Generates a title on the screen.
2)  printlogo(void);                Puts the software logo on the screen.
3)  load_file(void);                Lists all the STL files in the current
                                    directory and allows the user to open the
                                    desired file.
4)  display_rotate(void);           Displays a shaded image, along with the
                                    slicing plane and the intersection of the two.
                                :   Also rotates the shaded image and the
                                    slicing plane.
5)  enter_params(void);             The user inputs the slicing axis and
                                    thickness.
6)  is_op_allowable(char op);       Controls each procedure that is allowable
                                    to the user.
7)  menu(void);                     Outputs a menu.
8)  slice_and_display(void);        Slices the object and displays all slices.
9)  display_slice(void);            Displays each slice individually and then
                                    shows the layout of slices on each sheet.
10) make_hpgl_stuff(void);          Sorts the line file and converts it to HPGL
                                    file format.
11) print_all_pages(void);          Outputs the HPGL file to the plotter for
                                    cutting.
12) print_single_page(void);        Outputs one single sheet to the plotter for
                                    cutting.
13) clean_up(void);                 Deletes all of the intermediate files (.ln,
                                    .sr, and .hp files).

*****************************************************************************/
#include "menu.h"


int main_state;
int title(void);
void printlogo(void);
int load_file(void);
void display_rotate(void);
void enter_params(void);
int is_op_allowable(char op);
char menu(void);
void slice_and_display(void);
int make_hpgl_stuff(void);
void print_all_pages(void);
void print_single_page(void);
void clean_up(void);


char part_name[50], axis,c;
float thickness;
int  i,count_slice,showpage;
```

```
/*****************************************************************************
  This function generates a graphics title on the screen.
*****************************************************************************/
int title(void)
{
  int gdriver = DETECT, gmode, errorcode;
  int i,x,y,color,fill,xasp,yasp;
  char head[] = "Manufacturing lab";
  char title[] = "Shapemaker I ";
  char end[] = "press a key to continue";

  initgraph(&gdriver, &gmode,".");
   errorcode = graphresult();

  if (errorcode != grOk)  /* an error occurred */
  {
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1);              /* return with error code */
  }
  setbkcolor(BLUE);
  rectangle(0,0,getmaxx(),getmaxy());
  setcolor(YELLOW);
  settextjustify(LEFT_TEXT,TOP_TEXT);
  settextstyle(TRIPLEX_FONT,HORIZ_DIR,7);
  outtextxy((getmaxx()-textwidth(title))/2,getmaxy()/3,title);
  settextstyle(TRIPLEX_FONT,HORIZ_DIR,5);
  outtextxy((getmaxx()-textwidth(head))/2,getmaxy()/2,head);
  settextjustify(LEFT_TEXT,BOTTOM_TEXT);
  settextstyle(TRIPLEX_FONT,HORIZ_DIR,2);
  outtextxy((getmaxx()-textwidth(end))/2,getmaxy()-10,end);
  getch();
  closegraph();
  return 0;
}

/*****************************************************************************
  This function puts the software's logo on the screen.
*****************************************************************************/

void printlogo()
{
 clrscr();
 printf("\n");
 printf("-------------------------------------------------------------------\n")
 printf("\n");
 printf("    ZWSLICE 1.0\n");
 printf("\n");
 printf("-------------------------------------------------------------------\n")
 printf("\n");
 printf("\n");
 if(main_state<2)
   {
    printf("\n");
    printf("\n");
    printf("\n");
   }
 else
   {
```

```
    printf("      Part: %s\n",file_name);
    printf("      Axis: %c\n",axis);
    printf("      Slice Thickn ss: %1.3f mm\n",thickness);
    printf("      Slice Number: %d ",count_slice);
    printf("\n");
    printf("\n");
    printf("\n");
   }
}

/*****************************************************************************
  This function
  1)  Lists STL files in the current directory
  2)  Changes directory and driver
  3)    Loads files
*****************************************************************************/

int load_file()
{
  char ch;
  int  ins;
  clrscr();
  while(ch!='a' &&ch!='b')
  {
  clrscr();
  printf("\nPress a to list .stl file. Press b to enter file name. Press q to e:
  ch=getch();
  if(ch=='q' || ch=='Q')
  return 0;
  }
  switch(ch)
  {
  case 'a':
  case 'A':
  system("dir *.stl/w");

  case 'b':
  case 'B':
  ins=read_stl();
  if(ins==-1)
  return 0;
  object_size ();
  draw_stl_wire();
  printf("\n Object Size:\tx=%f(mm)\ty=%f(mm)\tz=%f(mm)\n",xmax, ymax, zmax);
  printf("\n");
  }
  printf("Press any key to continue");
  getch();
  main_state=1;
  return 0;
}
/*****************************************************************************
  This function
  1)    Displays: a shaded image with coordinate axes and slicing plane,
      along with the intersection of the two
  2)    Rotates:  a) the shaded image    b) the slicing plane
*****************************************************************************/
void display_rotate()
{
  return;
```

```
}

/*****************************************************************************
  This function allows the user to input slicing axis and thickness
*****************************************************************************/

void enter_params()
{
 char ch,tmpstring[80];

 clrscr();
 printlogo();

 axis=0;
 while(axis!='x' && axis!='y' && axis!='z')
   {
    printf("\nEnter axis normal to the slices (x,y, or z): ");
    scanf("%s",tmpstring);
    axis=tmpstring[0];
   }
 thickness=0.0;
 while(thickness<=0.0009 || thickness>=5.01)
   {

    printf("\nEnter the thickness of the slices (in mm): ");
    printf("\na------Paper thickness (0.123mm)");
    printf("\nb------Form  thickness (0.475mm)");
    printf("\nc------User  inpute ");
    printf("\n\n");
    ch=getche();
    switch(ch)
    {
      case  'a':
      case  'A':
      thickness=0.123;
      break;

      case 'b':
      case 'B':
      thickness=0.475;
      break;

      case 'c':
      case 'C':
      printf("\nEnter the thickness of the slices (in mm): ");
      scanf("%s",tmpstring);
      sscanf(tmpstring,"%f",&thickness);
      break;
    }
   }

 printf("\n\n");
 if(axis=='x')
 count_slice=(int)((xmax-xmin)/thickness);
 if(axis=='y')
 count_slice=(int)((ymax-ymin)/thickness);
 if(axis=='z')
 count_slice=(int)((zmax-zmin)/thickness);
 main_state=2;
}
```

```
/*****************************************************************************
   This function controls each procedure that is allowable to the user.
*****************************************************************************/

int is_op_allowable(char op)
{
 if(op=='1') return 1;
 if(op=='8') return 1;
 if(op=='9') return 1;
 if(op=='0') return 1;
 if(main_state>=1 && op=='2') return 1;
 if(main_state>=2 && op=='3') return 1;
 if(main_state>=3 && op=='4') return 1;
 if(main_state>=4 && op=='5') return 1;
 if(main_state>=5 && op=='6') return 1;
 if(main_state>=5 && op=='7') return 1;

 return 0;
}
/*****************************************************************************
  This function outputs an operation menu, hightlighting the
  operation that follows the last one completed.
*****************************************************************************/

char menu()
{
 char c;

 if(main_state==0){ textcolor(14);   cprintf("           1 - Load STL file  \r\
 if(main_state!=0){ textcolor(7);     printf("           1 - Load STL file  \n"
 if(main_state==1 && is_op_allowable('2')){textcolor(14);  cprintf("

 if(is_op_allowable('2') && main_state!=1){textcolor(7);    printf("

 if(main_state==2 && is_op_allowable('3')){textcolor(14);  cprintf("

 if(is_op_allowable('3') && main_state!=2){textcolor(7);    printf("

 if(main_state==3 && is_op_allowable('4')){textcolor(14);  cprintf("

 if(is_op_allowable('4') && main_state!=3){textcolor(7);  printf("            4

 if(main_state==4 && is_op_allowable('5')){textcolor(14);  cprintf("

 if(is_op_allowable('5') && main_state!=4){textcolor(7);    printf("

 if(main_state==5 && is_op_allowable('6')){textcolor(14);  cprintf("
 if(is_op_allowable('6') && main_state!=5){textcolor(7);    printf("

 if(is_op_allowable('7')) printf("           7 - Print a single page\n");
 textcolor(7);
 printf("           8 - DOS shell\n");
 printf("           9 - Clean up\n");
 printf("           0 - Quit\n");
 do
   {
    c=getch();
   }
 while(!is_op_allowable(c));
```

```
 return c;
}
/*****************************************************************************
  This function slices the object and generates a series of .ln files.
  It also displays s veral slices in 3D view.
*****************************************************************************/
void slice_and_display()
{

 int pages,showpage;

 clrscr();
 printf("Slicing the part...\n\n");
 if ( axis=='x')
 {
  slice_all(xmin,xmax,thickness);
  draw_all_slice(xmin,xmax,thickness);
 }
 else if ( axis=='y')
 {
  slice_all(ymin,ymax,thickness);
  draw_all_slice(ymin,ymax,thickness);
 }
 else if ( axis=='z')
 {
  slice_all(zmin,zmax,thickness);
  draw_all_slice(zmin,zmax,thickness);
 }


   printf("\n\n Done!  Press any key to continue");
  getch();
  main_state=3;
}
/*****************************************************************************
  This function
  1)  Displays each slice and the layout of slices on each sheet.
  2)  Flips some slices according to user input.
*****************************************************************************/

void display_slice()
{
  int pages;
  int gdriver = DETECT, gmode;
  clrscr();
  pages=count_pages();
  initgraph(&gdriver, &gmode, ".");
  setviewport(0,0,getmaxx(),getmaxy(),0);
  for(i=1;i<=number_of_slices;i++)
  {
    clearviewport();
    c=draw_slice(i);

    if(c=='q') break;

  }
  closegraph();
  printf("\nPress any key to display layout, Press  q to quit\n");
  getch();
  initgraph(&gdriver, &gmode, ".");
```

```
  setviewport(0,0,getmaxx(),getmaxy(),0);
  for(i=1;i<=pages;i++)
  {
    clearviewport();
    c=lay_out(i);
    if(c=='q') break;
  }
  closegraph();
  arrange_ment();
  flip_slice();
 printf("\n\n Done!  Press any key to continue");
 getch();
 main_state=4;
}
/*****************************************************************************
  This function lets the user choose which sheet to mirror.
  Each line file is sorted so that the lines connect at the ends,
  forming a continuous loop.  The line files are then converted into
  HPGL file format.
*****************************************************************************/

int make_hpgl_stuff()
{
 FILE *sizefile;
 int mirror,presee;
 char mirror_page,ch;
 mirror=0;
 clrscr();
 printf("  Do you want to mirror paper?\n");
 printf("  Enter Y for yes,  N  for  no.");
 ch=getch();
 switch(ch)
 {
    case 'Y':
    case 'y':
    mirror=1;
    break;

    case 'N':
    case 'n':
    mirror=0;
    break;

    case '\x1B':
    default:
    mirror=0;
    break;
 }
    if(mirror==1)
    {

     printf("\nEnter mirror choice\n");
     printf("  f-------first pager mirror\n");
     printf("  l-------last  pager mirror\n");
     printf("  Hit Esc  to escape\n");
     ch=getch();
     switch(ch)
     {
      case 'F':
      case 'f':
```

```
      mirror_page='f';
      break;

      case 'L':
      case 'l':
      mirror_page='l';
      break;

      case '\x1B':
      default:
      break;
     }
    }

 clrscr();
 printf("Sorting the lines...\n\n");
 presee=sort_line();
 if(presee==-1)
 return -1;

 printf("\n\nConverting data to hpgl format...\n\n");
  sizefile=fopen("objsize","w");
    if ( axis=='z')
    {
    fprintf(sizefile,"%f %f",xmax-xmin,ymax-ymin);
    }
    else if ( axis=='x')
    {
    fprintf(sizefile,"%f %f",ymax-ymin,zmax-zmin);
    }
    else if ( axis=='y')
    {
    fprintf(sizefile,"%f %f",xmax-xmin,zmax-zmin);
    }
  fclose(sizefile);
 creat_hpgl(mirror,mirror_page);
 printf("\n\n Done!  Press any key to continue");
 getch();
 main_state=5;
 return 0;
}
/*****************************************************************************
  This function outputs the HPGL files to the plotter, printing one
  page at a time.
*****************************************************************************/

void print_all_pages()
{
 char tmpstring[80],ch;
 int page;

 clrscr();
 printf("\n\nPrinting all %d pages...\n\n",Numpages);
 for(page=1;page<=Numpages;page++)
 {
    printf("\nInsert the  page  %d in the plotter and press p key...or press Es

    do
    { if ((ch=getch()) == '\x1B')
          break;
```

```
      else if(ch=='p')
      plot_out(page);
    }while (ch!='p');
 }
 printf("\n\n Done!  Press any key to continue");
 getch();
}
/*****************************************************************************
    This function only outputs one page of HPGL file to the plotter.
*****************************************************************************/
void print_single_page()
{
 char tmpstring[80],ch;
 int page;

 clrscr();
 page=0;
 while(page<1 || page>Numpages)
   {
    printf("\n\nEnter the page number to be printed (1-%d): ",Numpages);
    scanf("%s",tmpstring);
    sscanf(tmpstring,"%d",&page);
   }
 printf("Page %d will be plot\n", page);
 printf("Insert page in the plotter and press p...or press Esc to escape");
   do
    { if ((ch=getch()) == '\x1B')
         break;
      else if(ch=='p')
      plot_out(page);
    }while (ch!='\r');
 printf("\n\n Done!  Press any key to continue");
 getch();
}

/*****************************************************************************
    This function deletes all intermediate (.ln, .sr, and .hp) files.
*****************************************************************************/
void clean_up()
{
 printf("\n\nCleaning up intermediary files...\n\n");
 system("del *.hp");
 system("del *.ln");
 system("del *.sr");
 printf("\n\n Done!  Press any key to continue");
 main_state=1;
 getch();
}
/*****************************************************************************
  This is the main function and only works when the user inputs numbers
  between 0 to 9.  A function will be available only after the previous
  function has been completed successfully.
  If the user enters, 0 the software will quit.
*****************************************************************************/

void main()
{
 char option;

 main_state=0;
```

```
 title();
 do
   {

    printlogo();
    option=menu();
    if(option=='1') load_file();
    else if(option=='2') enter_params();
    else if (option=='3') slice_and_display();
    else if (option=='4') display_slice();
    else if (option=='5') make_hpgl_stuff();
    else if (option=='6') print_all_pages();
    else if (option=='7') print_single_page();
    else if (option=='8') dos_shell();
    else if (option=='9') clean_up();
    clrscr();
   }
 while (option!='0');
}
/*************************************************************************************
  End of program.
*************************************************************************************
```

```
/*****************************************************************************
  SEARCH1B.CPP
  Shapemaker 1 Library function
  This subprogram reads the STL file and slices the solid object in either
  the x, y, or z direction by finding the cross sections where the slice
  planes intersect the object datafile.
  This software is compiled and linked under Borlandc C++ 3.1 enviroment.
  Written by Zetian Wang

Function Description
1)  read_stl(void);              Reads the STL file from a CAD program.
                                 Saves the data file in x,y,z coordinates.
2)  object_size (void);          Finds the object size (maximum dimensions
                                 in the x, y, and z directions).
3)  slice_all (h_bot,h_top,dh);Finds the number of slices needed and
                                 generates the slice plane.
4)  get_line (slice_h,count);    Finds the intersection between the slice
                                 plane and the object.  Saves all line
                                 sections into .ln files (with x1, y1, x2,
                                 y2 format).
5)  draw_stl_wire(void);         Reads the STL file and generates a 3D
                                 wireframe image.
6)  draw_all_slice(h_bot,h_top,dh); Draws thick slices of line files in 3D
                                 view.
7)  draw_slice(int slice);       Outputs each slice and the layout of
                                 each sheet on the screen.
8)  arrange_ment(void);          Decomposes the object into thick slices
                                 and saves it to a data file.
9)  flip_slice(void);            Flips the slice according to user input.
10) dos_sheel(void);             Temporarily goes to a DOS enviroment,
                                 where all DOS commands can be executed.
11) regist_hole(float x_size); Finds the horizontal distance between
                                 registration holes for the object.
*****************************************************************************/
#include "menu.h"

  unsigned int Number;
  void object_size (void);
  void get_line (float slice_h,int count);
  float paper_size_width,paper_size_hight;
  float regist_hole(float x_size);
  void slice_all (float h_bot, float h_top, float dh);
  void draw_stl_wire(void);
  void draw_all_slice(float h_bot, float h_top, float dh);
  void flip_parts(void);
  char draw_slice(int slice);

  int number_of_slices,num_file;
  char infile_name[15], datafile_name[15], file_name[15],data_file_name[15];

/*****************************************************************************
  This function loads the STL file.  The units of measure needed for the
  object are input (inch or mm or user_scale).
  A data file (part.dat) is generated in x, y, z format.
*****************************************************************************/
int  read_stl(void)
{
    FILE *fptr;
    FILE *fdata;
    double x,y,z,unit;
```

```
  char temp[15],ch;
  char name1[] = "vertex";

/* attempt to open STL file for reading */

  file_name[0]=0; infile_name[0]=0; data_file_name[0]=0;
  printf("\nEnter the STL file name to open (Don't use file extension, wing f
  printf("\nor enter q to exit\n");
  scanf("%s",file_name);
  ch=file_name[0];
  if(ch=='q' || ch=='Q')
  return (-1);
  sprintf(infile_name,"%s.stl", file_name);
  sprintf(data_file_name,"%s.old", file_name);

  while ((fptr = fopen(infile_name, "r"))==NULL)
  {
        printf("\nCan't find %s file, please enter again\n", file_name );
        printf("\nor enter q to exit\n");
        file_name[0]=0; infile_name[0]=0; data_file_name[0]=0;
        scanf("%s",file_name);
        ch=file_name[0];
        if(ch=='q' || ch=='Q')
        return (-1);
        sprintf(infile_name,"%s.stl", file_name);
        sprintf(data_file_name,"%s.old", file_name);

  }

  printf("\nPlease enter object unit");
  printf("\ni:   inch        ");
  printf("\nm:   mm          ");
  printf("\nu:   user-scale  ");
  printf("\n\n");
  ch=getche();
  switch(ch)
  {
     case 'i':
     case 'I':
     unit=25.40;
     break;

     case 'm':
     case 'M':
     unit=1.0;
     break;

     case 'u':
     case 'U':
     printf("\nenter user-scale\n");
     scanf("%lf", &unit);
     break;

     default:
     unit=1.0;
     break;
}
  Number=0;
  fdata = fopen(data_file_name,"w");
  do
```

```
    {
          fscanf(fptr,"%s",temp);
          if (strcmp(temp,name1)==0)
          {
          fscanf(fptr,"%lf%lf%lf",&x,&y,&z);
          fprintf(fdata,"%.7g\t%.7g\t%.7g\n",x*unit,y*unit,z*unit);
          Number=Number+1;
          }

    } while (fgetc(fptr)!=EOF);


    printf("\n %d triangular find", Number/3);


    fclose(fdata);
    fclose(fptr);
    return 0;
}

/*****************************************************************************
  This function reads the data file (part.dat) and finds the object size
  (maximum dimensions in X, Y, and Z directions).  All the data will shift
  to positive coordinates with Xmin, Ymin, and Zmin reset to zero.
*****************************************************************************/
float xmin,xmax,ymin,ymax,zmin,zmax;
void   object_size()
{
  int i;
  float x,y,z,x_new,y_new,z_new,x_min,y_min,z_min,x_max,y_max,z_max;
  FILE *fdata, *sizefile, *data;
  xmin=50000; xmax=-100000;      x_min=50000; x_max=-100000;
  ymin=50000; ymax=-100000;      y_min=50000; y_max=-100000;
  zmin=50000; zmax=-100000;      z_min=50000; z_max=-100000;

  if  (( data = fopen(data_file_name,"r"))==NULL)
    printf("\n cannot open the file: %s\n",datafile_name);
  else
  {
    fscanf (data,"%f%f%f",&x,&y,&z);
    x_min=x;
    y_min=y;
    z_min=z;


    printf("\nFinding object size. Please wait...\n");
    for (i=0; fscanf(data,"%f%f%f",&x,&y,&z)!=EOF; i++)
    {
    if (x<x_min)
      x_min=x;
    if (x>x_max)
      x_max=x;
    if (y<y_min)
      y_min=y;
    if (y>y_max)
      y_max=y;
    if (z<z_min)
      z_min=z;
    if (z>z_max)
      z_max=z;
```

```
    }
    rewind(data);
    datafile_name[0]=0;
    sprintf(datafile_name,"%s.dat", file_name);
    if  ((data = fopen(data_file_name,"r"))==NULL)
    {
      printf("\n cannot open the file: %s\n",data_file_name);
      exit(-1);
    }
    fdata = fopen(datafile_name,"w");
    for (i=0; fscanf(data,"%f%f%f",&x,&y,&z)!=EOF; i++)
    {
       x_new=x-x_min;
       y_new=y-y_min;
       z_new=z-z_min;
       fprintf(fdata,"%.7g\t%.7g\t%.7g\n",x_new,y_new,z_new);

    }
       xmin=x_min-x_min;     xmax=x_max-x_min;
       ymin=y_min-y_min;     ymax=y_max-y_min;
       zmin=z_min-z_min;     zmax=z_max-z_min;


    sizefile=fopen("objsize","w");
    if ( axis=='z')
    {
    fprintf(sizefile,"%f %f",xmax-xmin,ymax-ymin);
    }
    if ( axis=='x')
    {
    fprintf(sizefile,"%f %f",ymax-ymin,zmax-zmin);
    }
    if ( axis=='y')
    {
    fprintf(sizefile,"%f %f",xmax-xmin,zmax-zmin);
    }
    fclose(sizefile);
    fclose(fdata);
    fclose(data);

  }
}

/*****************************************************************************
    This function finds the number of slices and generates the slice
    plane.
*****************************************************************************/
int count;
void slice_all(float h_bot, float h_top, float dh)
{

  int i;
  int num_slice;
  float slice_h;

  num_slice=(int)(( h_top - h_bot)/dh);

  number_of_slices=num_slice;
  num_file = num_slice ;
  printf(" The num_slice is %d",num_slice);
```

```
            if ( z3>0 )
            {

              n2=fabs(z2);
              n3=fabs(z3);
              x=(n2*x3+n3*x2)/(n2+n3);
              y=(n2*y3+n3*y2)/(n2+n3);
              fprintf(fline,"\n%.7g\t%.7g",x,y);

              n1=fabs(z1);
              n3=fabs(z3);
              x=(n1*x3+n3*x1)/(n1+n3);
              y=(n1*y3+n3*y1)/(n1+n3);
              fprintf(fline,"\t%.7g\t%.7g",x,y);

            }
          }

      }
    }
    }

    fclose (fdata);
    fclose (fline);
    return;
}
/*****************************************************************************
    This function reads the data file and generates a 3D wireframe
    picture.  The object is scaled and centered to fit the screen.
*****************************************************************************/
void draw_stl_wire(void)
{
    int i,poly[8],x,y;
    char ch;
    float x1,x2,x3,y1,y2,y3,z1,z2,z3,s,s1,s2;
    float dy,dz;
    int gdriver = DETECT, gmode, errorcode;
    FILE *fdata;

  /* initialize graphics mode */

    initgraph(&gdriver, &gmode, "C:\\borlandc\\bgi");

  /* read result of initialization */
    errorcode = graphresult();

    if (errorcode != grOk)  /* an error occurred */
    {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(-1);             /* return with error code */
    }


    if  (( fdata = fopen(datafile_name,"r"))==NULL)
      printf("\n cannot open the file: %s\n",datafile_name);
    else
    {
      dy=0.866*0.3;
```

```
  for ( i=1,count=1; i<=num_file; i++,count++ )
  {
     slice_h = (float)(h_bot + i*dh);

     get_line(slice_h,count);
     clrscr();
     printf("Slicing layer %d", i );
  }
}

/*****************************************************************************
    This function finds the intersection between data file and slice
    plane.  It saves all the intersections in line files (.ln) with line
    segement format (x1, y1, x2, y2).
*****************************************************************************/
void get_line(float slice_h,int count)

{
    int i,check;
    double x,y,x1,x2,x3,y1,y2,y3,z1,z2,z3;
    double n1,n2,n3;
    FILE *fdata;
    FILE *fline;
    char string[25];
    char  extension[] = ".ln";
    itoa(count, string,10);
    strcat (string,extension );


    fdata= fopen(datafile_name,"r");
    fline = fopen( string ,"w");

    for(i=0; i<Number/3;i++)
    {

      if (axis=='x')
      {
          fscanf (fdata,"%lf%lf%lf",&z1,&x1,&y1);
          fscanf (fdata,"%lf%lf%lf",&z2,&x2,&y2);
          fscanf (fdata,"%lf%lf%lf",&z3,&x3,&y3);
      }

      else if ( axis=='y')
      {
          fscanf (fdata,"%lf%lf%lf",&x1,&z1,&y1);
          fscanf (fdata,"%lf%lf%lf",&x2,&z2,&y2);
          fscanf (fdata,"%lf%lf%lf",&x3,&z3,&y3);
      }

      else if ( axis=='z')
      {
          fscanf (fdata,"%lf%lf%lf",&x1,&y1,&z1);
          fscanf (fdata,"%lf%lf%lf",&x2,&y2,&z2);
          fscanf (fdata,"%lf%lf%lf",&x3,&y3,&z3);

      }

     z1= z1 - slice_h;
     z2= z2 - slice_h;
```

```
  z3= z3 - slice_h;

  check =1;
  if(z1>0 && z2>0 && z3>0)
  check = 0;
  if(z1<0 && z2<0 && z3<0)
  check = 0;

  if (check==1)
  {
      if ( z1==0)
      {
/* 1)------------------check for condition of z1=0-------------- */
          if( z2 == 0 )
          {
             if ( z3 != 0 )
             fprintf(fline,"\n%.7g\t%.7g\t%.7g\t%.7g",x1,y1,x2,y2);
              }

         if ( z2 > 0 )
         {
             if ( z3<=0 )
             {
              n2=fabs(z2);
              n3=fabs(z3);
              x = (n2*x3+n3*x2)/(n2 + n3);
              y = (n2*y3+n3*y2)/(n2 + n3);
              fprintf(fline,"\n%.7g\t%.7g\t%.7g\t%.7g",x1,y1,x,y);
             }
          }

         if ( z2 < 0 )
         {
            if ( z3 >= 0)
            {
               n2=fabs(z2);
               n3=fabs(z3);
               x = (n2*x3+n3*x2)/(n2 + n3);
               y = (n2*y3+n3*y2)/(n2 + n3);
               fprintf(fline,"\n%.7g\t%.7g\t%.7g\t%.7g",x1,y1,x,y);
             }
         }
    }

/* 2)-----------------check for condition of z1 > 0------------*/
    if ( z1>0 )
    {
        if( z2==0 )
        {
          if (z3<=0 )
          {
            n1=fabs(z1);
            n3=fabs(z3);
            x=(n1*x3+n3*x1)/(n1+n3);
            y=(n1*y3+n3*y1)/(n1+n3);
            fprintf(fline,"\n%.7g\t%.7g\t%.7g\t%.7g",x2,y2,x,y);
          }
        }

        if( z2>0 )
```

```
    {
      if ( z3<0 )
      {
        n1=fabs(z1);
        n3=fabs(z3);
        x=(n1*x3+n3*x1)/(n1+n3);
        y=(n1*y3+n3*y1)/(n1+n3);
        fprintf(fline,"\n%.7g\t%.7g",x,y);

        n2=fabs(z2);
        n3=fabs(z3);
        x=(n2*x3+n3*x2)/(n2+n3);
        y=(n2*y3+n3*y2)/(n2+n3);
        fprintf(fline,"\t%.7g\t%.7g",x,y);
      }
    }

    if( z2<0 )
    {
      if ( z3==0 )
      {
        fprintf(fline,"\n%.7g\t%.7g",x3,y3);
        n1=fabs(z1);
        n2=fabs(z2);
        x=(n1*x2+n2*x1)/(n1+n2);
        y=(n1*y2+n2*y1)/(n1+n2);
        fprintf(fline,"\t%.7g\t%.7g",x,y);
      }

      else if ( z3>0 )
      {
        n1=fabs(z1);
        n2=fabs(z2);
        x=(n1*x2+n2*x1)/(n1+n2);
        y=(n1*y2+n2*y1)/(n1+n2);
        fprintf(fline,"\n%.7g\t%.7g",x,y);

        n2=fabs(z2);
        n3=fabs(z3);
        x=(n2*x3+n3*x2)/(n2+n3);
        y=(n2*y3+n3*y2)/(n2+n3);
        fprintf(fline,"\t%.7g\t%.7g",x,y);
      }
      else
      {
        n1=fabs(z1);
        n2=fabs(z2);
        x=(n1*x2+n2*x1)/(n1+n2);
        y=(n1*y2+n2*y1)/(n1+n2);
        fprintf(fline,"\n%.7g\t%.7g",x,y);

        n1=fabs(z1);
        n3=fabs(z3);
        x=(n1*x3+n3*x1)/(n1+n3);
        y=(n1*y3+n3*y1)/(n1+n3);
        fprintf(fline,"\t%.7g\t%.7g",x,y);
        }
    }
}
```

```
/* 3)-------------------- check for condition of z1<0 ---------------*/
    if ( z1<0 )
    {
        if ( z2==0 )
        {
           if (z3 >= 0 )
           {
             fprintf(fline,"\n%.7g\t%.7g",x2,y2);
             n1=fabs(z1);
             n3=fabs(z3);
             x=(n1*x3+n3*x1)/(n1+n3);
             y=(n1*y3+n3*y1)/(n1+n3);
             fprintf(fline,"\t%.7g\t%.7g",x,y);
           }
        }

        if ( z2>0 )
        {
           if ( z3==0 )
           {
             fprintf(fline,"\n%.7g\t%.7g",x3,y3);
             n1=fabs(z1);
             n2=fabs(z2);
             x=(n1*x2+n2*x1)/(n1+n2);
             y=(n1*y2+n2*y1)/(n1+n2);
             fprintf(fline,"\t%.7g\t%.7g",x,y);
           }

           else if ( z3>0 )
           {
             n1=fabs(z1);
             n2=fabs(z2);
             x=(n1*x2+n2*x1)/(n1+n2);
             y=(n1*y2+n2*y1)/(n1+n2);
             fprintf(fline,"\n%.7g\t%.7g",x,y);

             n1=fabs(z1);
             n3=fabs(z3);
             x=(n1*x3+n3*x1)/(n1+n3);
             y=(n1*y3+n3*y1)/(n1+n3);
             fprintf(fline,"\t%.7g\t%.7g",x,y);
           }
           else
           {
             n1=fabs(z1);
             n2=fabs(z2);
             x=(n1*x2+n2*x1)/(n1+n2);
             y=(n1*y2+n2*y1)/(n1+n2);
             fprintf(fline,"\n%.7g\t%.7g",x,y);

             n2=fabs(z2);
             n3=fabs(z3);
             x=(n2*x3+n3*x2)/(n2+n3);
             y=(n2*y3+n3*y2)/(n2+n3);
             fprintf(fline,"\t%.7g\t%.7g",x,y);
           }
        }

        if ( z2<0 )
        {
```

```
dz=0.5*0.3;
s1=0.0;
s2=0.0;
s=0.0;
if((ymax-ymin) >= (zmax-zmin ) && (ymax-ymin) >= (xmax-xmin))
{
  s1= (getmaxx()/(ymax-ymin));
  s2= (getmaxy()/(zmax-zmin));
  if(s1>=s2)
  s = s2*0.85;
  else
  s = s1*0.85;
}
else if((zmax-zmin) >= (ymax-ymin) && (zmax-zmin) >= (xmax-xmin))
{
  s1= (getmaxx()/(ymax-ymin));
  s2= (getmaxy()/(zmax-zmin));
  if(s1>=s2)
  s = s2*0.85;
  else
  s = s1*0.85;

}
else if((xmax-xmin) >= (ymax-ymin) && (xmax-xmin) >= (zmax-zmin))
{
  s1= (getmaxx()/(xmax-xmin));
  s2= (getmaxy()/(ymax-ymin));
  if(s1>=s2)
  s = s2*0.85;
  else
  s = s1*0.85;

}
setviewport(0,0,getmaxx(),getmaxy(),0);
setbkcolor(1);
setcolor(14);
if (((xmax-xmin) <= (zmax-zmin)) || ((xmax-xmin) <=(ymax-ymin)))
{
  for (i=1; i<Number/3 ;++i)
  {
    fscanf(fdata,"%f%f%f",&x1,&y1,&z1);
    fscanf(fdata,"%f%f%f",&x2,&y2,&z2);
    fscanf(fdata,"%f%f%f",&x3,&y3,&z3);

    y1= (s*((y1)-(x1-xmin)*dy - (ymax+ymin)/2) + (getmaxx()/2)) ;
    y2= (s*((y2)-(x2-xmin)*dy - (ymax+ymin)/2) + (getmaxx()/2)) ;
    y3= (s*((y3)-(x3-xmin)*dy - (ymax+ymin)/2) + (getmaxx()/2)) ;
    z1= (s*((z1)-(x1-xmin)*dz - (zmax+zmin)/2) + (getmaxy()/2)) ;
    z2= (s*((z2)-(x2-xmin)*dz - (zmax+zmin)/2) + (getmaxy()/2)) ;
    z3= (s*((z3)-(x3-xmin)*dz - (zmax+zmin)/2) + (getmaxy()/2)) ;

    line(y1,z1,y2,z2);

    line(y2,z2,y3,z3);

    line(y3,z3,y1,z1);
  }
}
else
{
```

```
        for (i=1; i<Number/3 ;++i)
        {
          fscanf(fdata,"%f%f%f",&x1,&y1,&z1);
          fscanf(fdata,"%f%f%f",&x2,&y2,&z2);
          fscanf(fdata,"%f%f%f",&x3,&y3,&z3);


          x1= (s*((x1)-(z1-zmin)*dy - (xmax+xmin)/2) + (getmaxx()/2)) ;
          x2= (s*((x2)-(z2-zmin)*dy - (xmax+xmin)/2) + (getmaxx()/2)) ;
          x3= (s*((x3)-(z3-zmin)*dy - (xmax+xmin)/2) + (getmaxx()/2)) ;
          y1= (s*((y1)-(z1-zmin)*dz - (ymax+ymin)/2) + (getmaxy()/2)) ;
          y2= (s*((y2)-(z2-zmin)*dz - (ymax+ymin)/2) + (getmaxy()/2)) ;
          y3= (s*((y3)-(z3-zmin)*dz - (ymax+ymin)/2) + (getmaxy()/2)) ;


          line(x1,y1,x2,y2);

          line(x2,y2,x3,y3);

          line(x3,y3,x1,y1);
        }

      }

    }

      /* clean up */

    getch();
    closegraph();

    return ;

}
/*****************************************************************************
    This function reads the line files and outputs several line
    files in 3D format.  The entire picture is scaled and shifted to the
    center of the screen.
*****************************************************************************/

void draw_all_slice( float h_bot, float h_top, float dh )
{
  int i,j,num;
  float x1,x2,y1,y2,z1,z2,s1,s2,s,y;
  float dy,dz,h;
  float xl,xr,yt,yb,sx,sy,snew,delt_x,delt_y;
  int gdriver = DETECT, gmode;
  FILE *fline;
  char string[35];
  char  extension[] = ".ln";
  /* initialize graphics mode */

  initgraph(&gdriver, &gmode, "C:\\borlandc\\bgi");


  dy=0.866*0.3;
  dz=0.5*0.3;
  if ( axis=='z')
  {
```

```
  if(zmax>xmax && zmax>ymax)
  s=getmaxy()/(zmax-zmin);
  else
  {
    s1 = (getmaxy()/(ymax - ymin));
    s2 = (getmaxy()/(xmax - xmin));
    if (s1 >=s2)
    s=s2;
    else
    s=s1;
  }
}
else if ( axis=='x')
{
  if(xmax>ymax && xmax>zmax)
  s=getmaxy()/(xmax-xmin);
  else
  {
    s1 = (getmaxy()/(ymax - ymin));
    s2 = (getmaxy()/(zmax - zmin));
    if (s1 >=s2)
    s=s2;
    else
    s=s1;
  }
}
else if ( axis=='y')
{
  if(ymax>xmax && ymax>zmax)
  s=getmaxy()/(ymax-ymin);
  else
  {
    s1 = (getmaxy()/(xmax - xmin));
    s2 = (getmaxy()/(zmax - zmin));
    if (s1 >=s2)
    s=s2;
    else
    s=s1;
  }
}

setviewport(0,0,getmaxx(),getmaxy(),0);
setbkcolor(1);
setcolor(14);

if(num_file<=10)
num=1;
else
num= ceil(num_file/10);
xl=25000.0; xr=-25000.0;
yt=25000.0; yb=-25000.0;
for ( i=1;i<=num_file; i+=num)
{
  h=(float)(h_bot+i*dh);
  itoa(i, string,10);
  strcat (string,extension );
  fline = fopen( string ,"r");
  for ( j=1 ; j<=Number && fgetc(fline)!=EOF ; j++)
   {
    fscanf(fline,"%f%f%f%f",&x1,&y1,&x2,&y2);
```

```
      y1= (s*((y1)-(x1-xmin)*dy - (ymax+ymin)/2) + (getmaxx()/2)) ;
      y2= (s*((y2)-(x2-xmin)*dy - (ymax+ymin)/2) + (getmaxx()/2)) ;
      z1= (s*((h)-(x1-xmin)*dz - (h_top+h_bot)/2) + (getmaxy()/2)) ;
      z2= (s*((h)-(x2-xmin)*dz - (h_top+h_bot)/2) + (getmaxy()/2)) ;
      if(xl>y1)  xl=y1;     if(xl>y2)  xl=y2;
      if(xr<y1)  xr=y1;     if(xr<y1)  xr=y1;
      if(yt>z1)  yt=z1;     if(yt>z2)  yt=z2;
      if(yb<z1)  yb=z1;   if(yb<z2)  yb=z2;

     }
    fclose(fline);
  }

    sx=(xr-xl)/getmaxx();
    sy=(yb-yt)/getmaxy();

    if(sx<=1.0 && sy<=1.0)
    {
      snew=1.0;
      delt_x=(getmaxx()-(xr+xl))/2.0 ;
      delt_y=(getmaxy()-(yb+yt))/2.0 ;
    }
    else
    {
      snew=((sx > sy) ? sx : sy);
      snew=1.0/snew;
      delt_x=(getmaxx()-(xr+xl))/2.0 ;
      delt_y=(getmaxy()-(yb+yt))/2.0 ;

    }

  for ( i=1;i<=num_file; i+=num)
  {
    h=(float)(h_bot+i*dh);
    itoa(i, string,10);
    strcat (string,extension );
    fline = fopen( string ,"r");
    for ( j=1 ; j<=Number && fgetc(fline)!=EOF ; j++)
     {
      fscanf(fline,"%f%f%f%f",&x1,&y1,&x2,&y2);
      y1= ((s*(y1-(x1-xmin)*dy - (ymax+ymin)/2) + getmaxx()/2) + delt_x)*snew;
      y2= ((s*(y2-(x2-xmin)*dy - (ymax+ymin)/2) + getmaxx()/2) + delt_x)*snew ;
      z1= ((s*(h -(x1-xmin)*dz - (h_top+h_bot)/2) + getmaxy()/2) +delt_y)*snew ;
      z2= ((s*(h -(x2-xmin)*dz - (h_top+h_bot)/2) + getmaxy()/2) +delt_y)*snew ;


      line(y1,z1,y2,z2);
    }
    fclose(fline);
  }
       /* clean up */
  getch();
  closegraph();
  return ;
}
/*****************************************************************************
  This function draws single slices and single pages of layout.  The last
  page will show the registration holes.  All the images are scaled and
  centered to fit the screen.
*****************************************************************************/
```

```
char draw_slice(int slice)
{
  int i;
  float x1,x2,xa,xb,y1,y2,ya,yb,z1,z2,za,zb,s1,s2,s;
  float dy,dz;
  FILE *fdata;
  char slicefilename[15],c,text[15];

  sprintf(text,"Slicing %d",slice);
  settextstyle(TRIPLEX_FONT,HORIZ_DIR,2);
  outtextxy((getmaxx()-textwidth(text))/2,15,text);
  if ( axis=='z')
  {
    if((ymax-ymin) >= (xmax - xmin))
    {
      s = (getmaxy()/(ymax - ymin))*0.7;
    }
    else if((ymax-ymin) < (xmax - xmin))
    {
      s = (getmaxx()/(xmax - xmin))*0.7;
    }
  }
  else if ( axis=='x')
  {
      s1=(getmaxx()/(ymax-ymin));
      s2=(getmaxy()/(zmax-zmin));
      if(s1>=s2)
      s=s2*0.9;
      else
      s=s1*0.9;
  }
  else if ( axis=='y')
  {
    if((xmax-xmin) >= (zmax - zmin))
    {
      s = (getmaxx()/(xmax - xmin))*0.7;
    }
    else if((xmax-xmin) < (zmax - zmin))
    {
      s = (getmaxy()/(zmax - zmin))*0.7;
    }
  }


  sprintf(slicefilename,"%d.ln",slice);
  if  (( fdata = fopen(slicefilename,"r"))==NULL)
  printf("\n cannot open the file: %s\n",slicefilename);
  else
  {
    setbkcolor(1);
    setcolor(14);
    for (i=1;  fgetc(fdata)!=EOF;i++)
    {
      if ( axis=='z')
      {
        fscanf(fdata,"%f%f%f%f",&x1,&y1,&x2,&y2);
        xa=(s*(x1-(xmax+xmin)/2) + getmaxx()/2);
        xb=(s*(x2-(xmax+xmin)/2) + getmaxx()/2);
        ya=(s*(y1-(ymax+ymin)/2) + getmaxy()/2);
        yb=(s*(y2-(ymax+ymin)/2) + getmaxy()/2);
```

```
        line(xa,ya,xb,yb);

      }

      else if ( axis=='x')
      {
        fscanf(fdata,"%f%f%f%f",&y1,&z1,&y2,&z2);
        ya=(s*(y1-(ymax+ymin)/2) + getmaxx()/2);
        yb=(s*(y2-(ymax+ymin)/2) + getmaxx()/2);
        za=(s*(z1-(zmax+zmin)/2) + getmaxy()/2);
        zb=(s*(z2-(zmax+zmin)/2) + getmaxy()/2);
        line(ya,za,yb,zb);

      }

      else if ( axis=='y')
      {
        fscanf(fdata,"%f%f%f%f",&x1,&z1,&x2,&z2);
        xa=(s*(x1-(xmax+xmin)/2) + getmaxx()/2);
        xb=(s*(x2-(xmax+xmin)/2) + getmaxx()/2);
        za=(s*(z1-(zmax+zmin)/2) + getmaxy()/2);
        zb=(s*(z2-(zmax+zmin)/2) + getmaxy()/2);
        line(xa,za,xb,zb);
      }
    }

  }

  c=getch();
  fclose(fdata);
  return c;
}
/*****************************************************************************
    This function asks the user to input paper size (in landscopy format).
    It then caculates how many columns and rows of slices can fit on a
    page, and how many sheets are needed to build a part.  The slices are
    evenly arranged on the sheet.  The space between each slice is also
    calculated.  Presently we have only put a maximum of eight or nine
    slices on one sheet.
*****************************************************************************/

float x_shift, y_shift,x_size,y_size,x_offset,y_offset,s,xoff;
float x_size_with_hole;
int n=0,row,column,cond;
int count_pages(void)
{
  int   row2=0,column2=0,n2=0;
  float s1,s2;
  float net_wide, net_hight;
  char ch;
  if ( axis=='z')
  {
    x_size=xmax-xmin; y_size=ymax-ymin;
  }
  else if ( axis=='x')
  {
    x_size=ymax-ymin; y_size=zmax-zmin;
  }
  else if ( axis=='y')
  {
```

```
  x_size=xmax-xmin; y_size=zmax-zmin;
}

printf("\nPlease enter the paper size");
printf("\na:   width=11(in)          height=8.5(in)");
printf("\nb:   width=12(in)          height=9 (in)");
printf("\nc:   other size\n");
ch=getche();
switch(ch)
{
     case 'a':
     case 'A':
     paper_size_width=280.0;
     paper_size_hight=215.0;
     break;

     case 'b':
     case 'B':
     paper_size_width=300.0;
     paper_size_hight=230.0;
     break;

     case 'c':
     case 'C':
     printf("\nenter paper width in inch)\n");
     scanf("%f", &paper_size_width);
     paper_size_width=25.4*paper_size_width;
     printf("\n enter paper height in inch\n");
     scanf("%f", &paper_size_hight);
     paper_size_hight=25.4*paper_size_hight;
     break;

     default:
     paper_size_width=280.0;
     paper_size_hight=215.0;
     break;
}

s=0.0;
s1=getmaxx()/paper_size_width;
s2=getmaxy()/paper_size_hight;
row=0;   x_shift=0;
column=0;  y_shift=0;
if (s1 >= s2)
{
  s=s2;
}
else
  s=s1;

cond=debug;
if(cond==0)
{
xoff=10.0;
net_wide=paper_size_width - paper_side*2.0 ;
net_hight=paper_size_hight - paper_bot - paper_top;
x_size_with_hole=regist_hole(x_size);
row2=(int)(net_wide /(x_size_with_hole+xoff));
column2=(int)(net_hight /(y_size + offset));
```

```
if ((row2 && column2) >=1)
{
  if (row2*column2 >= 8)
  {
    if ((num_file%8)==0 )
    {
      n=(int)(num_file/8);
    }
    if((num_file%8)!=0)
    {
      n = (int)((num_file/(8))+1);
    }

    if((row2 >= 2) && (column2 >= 4))
    {
      row = 2;
      column = 4;
      x_offset = ( net_wide - 2.0*(x_size_with_hole + xoff));
      y_offset = ( net_hight - 4.0*y_size )/3.0;
    }
    else if((row2 >= 4) && (column2 >= 2))
    {
      row = 4;
      column = 2;
      x_offset = ( net_wide - 4.0*(x_size_with_hole + xoff))/3.0 ;
      y_offset = ( net_hight - 2.0*y_size );
    }
    else if((row2 >=1) && (column2 >=8))
    {
      row = 1;
      column = 8;
      x_offset = ( net_wide - 1.0*(x_size_with_hole + xoff)) ;
      y_offset = ( net_hight - 8.0*y_size )/7.0;
      x_shift = x_offset/2.0;
    }

    else if((row2 >=8) && (column2 >=1))
    {
      row = 8;
      column = 1;
      x_offset = ( net_wide - 8.0*(x_size_with_hole +xoff))/7.0 ;
      y_offset = ( net_hight - 1.0*y_size );
      y_shift = y_offset/2.0;


    }
    else if((row2 >=3) && (column2 >=3))
    {
      row = 3;
      column = 3;
      x_offset = ( net_wide - 3.0*(x_size_with_hole +xoff))/2.0 ;
      y_offset = ( net_hight - 3.0*y_size )/2.0;
      if ((num_file%9)==0 )
      {
        n=(int)(num_file/9);
      }
      if((num_file%9)!=0)
      {
        n = (int)((num_file/(9))+1)
```

```
        }
      }

    }
    if (row2*column2 < 8 )
    {
      if((num_file%(row2*column2))==0)
      {
        n2=(int)(num_file/(row2*column2));
      }
      if((num_file%(row2*column2))!=0)
      {
        n2=(int)((num_file/(row2*column2))+1);
      }
    n=n2;
    row = row2;
    column = column2;

    if((row>1) && (column>1))
    {
      x_offset = ( net_wide - row*(x_size_with_hole+xoff )) /(row - 1);
      y_offset = ( net_hight - column*y_size )/(column - 1);
    }
    if((row>1) && (column==1))
    {
      x_offset = ( net_wide - row*(x_size_with_hole+xoff)) /(row - 1);
      y_offset = ( net_hight - column*y_size )/(column );
      y_shift = y_offset/2.0;
    }
    if((row==1) && (column>1))
    {
      x_offset = ( net_wide - row*(x_size_with_hole+xoff)) /(row );
      y_offset = ( net_hight - column*y_size )/(column - 1);
      x_shift = x_offset/2.0;
    }
    }

  printf("\nEach sheet will contain  %d slices,%d rows and %d columns", row*colu
  printf("\n%d pages are needed to construct the part",n);
  }
  else
  printf ("\nPrototype size is too big for landscape\n");
  }   /*  end of debug==0  */

  printf("\n Press any key to continue");
  getch();
  Numpages=n;
  return n;

}
/*****************************************************************************
    This function records the decomposion information into a data file.
    A object is decomposed into thick slices.
*****************************************************************************/
void  arrange_ment()
{
  int i,j,n,fliping;
  char flip;
  FILE *file;
```

```
   if((file=fopen("slice.dat","w"))==NULL)
   printf("\nUnable to open file to write");

   for(i=1;i<=num_file; i+=Numpages)
   {
     n=1;
     for(j=i; n<=1;j+=Numpages)
     {
     fprintf(file,"%d\t",j);
     if(j+Numpages >num_file)
     fprintf(file,"%d\n",num_file);
     else
     fprintf(file,"%d\n",j+Numpages-1);
     n++;
     }

   }
   fclose(file);
}
/*****************************************************************************
    This function finds out whether the user wants to flip any slices.
*****************************************************************************/
void flip_slice()
{
     char flip;

     printf("Do you want to fliping the slice\n");
     printf("Enter y for YES, n for NO\n");
     while(flip!='y' && flip!='n')
     {
     flip=getch();
     switch(flip)
     {
     case 'n':
     case 'N':
     break;

     case 'Y':
     case 'y':
     flip_parts();
     break;
     }
     }
}
/*****************************************************************************
    This function will flip the slices according to user input.
*****************************************************************************/
void flip_parts()
{
    FILE *fp1;
    FILE *fp,*fp2;
    int i,j,m,n,mn;
    int limit;
    int data0[20],data1[20],data2[20];
    char flip,c,oldfile[20],temfile[20],input[80];
    int temp1[20];

     if((fp=fopen("slice.dat","r"))==NULL)
      printf("\nUnable to open file to read");
     i=0;
```

```
        do
        {
         fscanf(fp, "%d  %d", &data1[i], &data2[i]);
         i++;
        } while(fgetc(fp)!=EOF);
        fclose(fp);
        limit=i-1;
        printf("\nStack Number ");
        for(j=0;j<i-1; j++)
        {
         data0[j]=j;
         printf("\n%d): The slice is from %d to %d\n",data0[j], data1[j],data2[j])
        }
        printf("\nEnter stack number to flip with space betiween\n");
        gets(input);
        printf("the string input is: %s\n",input);
        if((fp1=fopen("slice.tmp","w"))==NULL)
        printf("Unable to open slice.tmp to write");
        printf("\nruning after get string");
        fprintf(fp1,"%s",input);
        fclose(fp1);

        if((fp2=fopen("slice.tmp","r"))==NULL)
        printf("\nUnable to open slice.tmp to read");
        rewind(fp2);
        for(i=0;fscanf(fp2,"%d",&temp1[i])!=EOF;i++)
        printf("\ntemp[%d]=%d",i,temp1[i]);
        fclose(fp2);

        for(j=0;j<limit;j++)
        {
         for(int k=0;k<i;k++)
         {

           if(data0[j]==temp1[k])
           {
             for( n=data1[j]; n<=data2[j];n++)
             {
              sprintf(oldfile,"%d.ln",n);
              sprintf(temfile,"%d.temp",n);
              rename(oldfile,temfile);
              printf("\nslice n=%d",n);
             }
             for( m=data1[j],n=0;m<=data2[j];m++,n++)
             {
              mn=m+data2[j]-m-n;
              sprintf(oldfile,"%d.temp",m);
              sprintf(temfile,"%d.ln",mn);
              rename(oldfile,temfile);
              printf("\n slice m=%d mn=%d",m,mn);
             }

           }

         }
       }
    }

/*****************************************************************************
    This function draws the layout of each sheet on the screen. On the
```

```
   last page, it also shows the registration holes.
*****************************************************************************
char lay_out(int page)
{
  int l=0,m=0,j=0,k=0,na,xc1,xc2,yc;
  FILE *fline;
  float x1,x2,y1,y2,xa,xb,ya,yb;
  char  string[15],c,text[15];
  char  extension[] = ".ln";
     cleardevice();
     setbkcolor(1);
     setcolor(14);
     rectangle(0,0,paper_size_width*s,paper_size_hight*s);
     sprintf(text,"page %d",page);
     settextstyle(TRIPLEX_FONT,HORIZ_DIR,2);
     outtextxy((getmaxx()-textwidth(text))/2,15,text);
     j=page;
     k=1; l=0;
     na=n;
     for ( m=j; m<=num_file  ; m+=na)      //&& k<=(column/2)//
     {
      l++;
      if (l== (column+1))
      {
        l=1;
        k++;
      }
      itoa(m, string,10);
      strcat (string,extension );

      if ((fline = fopen(string,"r"))==NULL)
      {
        printf("\n cannot open the file %s",string);
        break;
      }

     while(fscanf(fline,"%f%f%f%f", &x1,&y1,&x2,&y2)!=EOF)
     {

       xa=(int)((k-1)*(x_size_with_hole+x_offset+xoff)*s+(x1+x_shift+paper_side
       xb=(int)((k-1)*(x_size_with_hole+x_offset+xoff)*s+(x2+x_shift+paper_side

       ya=(int)((l-1)*(y_size+y_offset)*s+(y1-ymin+y_shift+paper_bot)*s );
       yb=(int)((l-1)*(y_size+y_offset)*s+(y2-ymin+y_shift+paper_bot)*s );

       line(xa,ya,xb,yb);

     }
     if(cond==0)
     {
     if(j==n)
     {
       xc1=(int)((k-1)*(x_size_with_hole+x_offset+xoff)*s+(paper_side+xoff/2.0+
       xc2=(int)(xc1+x_size_with_hole*s);
       yc =(int)((l-1)*(y_size+y_offset)*s+(y_size/2.0+y_shift+paper_bot)*s );
       /* draw the registration circle */
       circle(xc1, yc, 3.175*s);
       circle(xc2, yc, 3.175*s);
     }
     }
```

```
      rewind(fline);
      fclose(flin );

      }
      c=getch();
    return c;

}
/*****************************************************************************
    This function lets the user go to the dos shell to execute DOS
    commands.  Entering "exit" will return the user to the program.
*****************************************************************************/
void dos_shell(void)
{
   printf("Press EXIT to return...");
   /* go into the dos shell and enter exit to return */
   system(getenv("COMSPEC"));

}
/*****************************************************************************
    This function will find the distance between registration holes for
    different object sizes.
*****************************************************************************/
float regist_hole(float x_size)
{
   int i;
   float hole_dist;
   float array[32]={10.0,15.0,20.0,25.0,30.0,35.0,40.0,45.0,50.0,55.0,
   60.0,65.0,70.0,75.0,80.0,85.0,90.0,95.0,100.0,105.0,110.0,
   115.0,120.0,125.0,130.0,140.0,145.0,155.0,170.0,175.0,185.0,205.0};
   if(cond==0)
   {
     for(i=0;i<=31;i++)
     {
        if(array[i] > x_size)
        {

          hole_dist=array[i+2];
          break;
        }
        else if(array[31] <= x_size)
        {
          printf("\n  WARNING :  This part is too big to fit registration holes
        }
     }
      printf("\nthe hole_dist is %f", hole_dist);
   }
   else if(cond==1)
   {
     hole_dist=x_size;
   }
   getch();
  return hole_dist;
}
```

```
/*****************************************************************************
  Jassort.cpp
  Shapemaker 1 Library function.
  This subprogram reads line files (.ln) and sorts them into a sequenced
  line segement until the entire loop has been sorted and connected. If
  a double line is found, one of the two lines will be erased.
  After sorting, one sort file(.sr) will be generated in the line segment
  format (x1,y1,x2,y2).
  To save the memory of the system a dynamic array has been used to store
  each line file.
*****************************************************************************/
#include "menu.h"

int sort_line(void)
{

   unsigned int i,n,j,k=0,check,line_count;
   int  count,m;
   FILE *fline,*fline1;
   double x1,x2,y1,y2;
   char string[25];
   char string1[25];
   char  extension[] = ".ln";
   char  extension1[] =".sr";

   for ( m=1;m<=number_of_slices; m++)
   {
    count=0;
    itoa(m, string,10);
    itoa(m, string1,10);
    strcat (string,extension );
    strcat (string1,extension1);
    fline1= fopen( string1,"w");

    if ((fline = fopen(string,"r"))==NULL)
    {
      printf("\n Can't find line file. Slicing object again!!");
      printf("\n Press any key to continue");
      getch();
      return -1;
    }
    else
    {
      for (i=0;fscanf(fline,"%lf%lf%lf%lf",&x1,&y1,&x2,&y2)!=EOF ; i++) ;
      rewind(fline);
      n=i+1;
      double ( *coord)[4] = new double [n][4];
      if(coord==0)
      {
          printf("Run out of memory");
          exit(1);
      }
      for (i=0;fscanf(fline,"%lf%lf%lf%lf",&coord[i][0],&coord[i][1],&coord[i][2

      for (k=0; k<i;k++)
      {
        for(j=k+1; j<=i; j++)
        {
          if ((coord[k][0]==coord[j][0] && coord[k][1]==coord[j][1]) && (coord[k
          {
```

```
            printf("found double line 1");
            coord[j][0]=-1.0;
            count++;
         }
         else if ((coord[k][0]==coord[j][2] && coord[k][1]==coord[j][3]) && (c
         {
            printf("found double line 2");
            coord[j][0]=-1.0;
            count++;
         }
      }
}
x1= coord[0][0];x2=coord[0][2];
y1= coord[0][1];y2=coord[0][3];
line_count=i;
fprintf(fline1,"%lf\t%lf\t%lf\t%lf\n",x1,y1,x2,y2);
count++;

do {
  check = 0;
  // test if two lines are connected to each other//
  for (j=1; x1!=x2 || y1!=y2 ;j++)
  {

    if ( coord[j][0]!=-1.0)
    {

      if ( x2==coord[j][0] && y2==coord[j][1] )
      {
        fprintf(fline1,"%lf\t%lf\t%lf\t%lf\n",coord[j][0],coord[j][1],coord[
        x2 = coord[j][2];
        y2 = coord[j][3];
        coord[j][0]=-1.0;
        check =1;
        count++;


      }
      else if ( x2==coord[j][2] && y2==coord[j][3] )
      {
        fprintf(fline1,"%lf\t%lf\t%lf\t%lf\n",coord[j][2],coord[j][3],coord[
        x2 = coord[j][0];
        y2 = coord[j][1];
        coord[j][0]=-1.0 ;
        check =1;
        count++;
      }
    }
   if (j==line_count-1)
   j=0;
   }
   if (check==0)
    printf("Problem, No connected line at point %f   %f", x2,y2);
  else if (check==1)
  {
    fprintf(fline1,"\nloop has been connected\n");
  }

  if ( count<line_count )
  {
```

```
        for (n=1;n<=line_count; n++)
        {
          if ( coord[n][0]!=-1.0)
          {
            x1=coord[n][0];  x2=coord[n][2];
            y1=coord[n][1];  y2=coord[n][3];
            fprintf(fline1,"%lf\t%lf\t%lf\t%lf\n",x1,y1,x2,y2);
            coord[n][0]=-1.0;
            count++;
            break;
          }
        }

      }
    }
    while(count<line_count);

    fclose(fline1);
    fclose(fline);
    delete []coord;
  }
  clrscr();
  printf("\nfile %d has been sorted",m);

}
  return 0;
}
```

```
/*****************************************************************************
  HPGL4A.CPP
  Shapemaker 1 Library function
  This subprogram reads the sorted line files and convert them into HPGL
  file format. One HPGL file contains all of the slices that fit onto one
  sh et. The registration holes will be cut three times and only cut on
  the mirrored page.
  Created by Zetian Wang.
*****************************************************************************/
#include "menu.h"

int Numpages;

void creat_hpgl(int mirror,char mirror_page)
{
   int   i, m,l, count_line=0,cond;
   int   j,k,hole_page;
   int   x1i,x2i,y1i,y2i,xc1,xc2,yc;
   float xc,xoff;
   FILE  *fline1,*fhpgl,*filesize ;
   double  x1,y1,x2,y2;
   char  temp1[15],temp2[15],temp3[15],temp4[15];
   char  xc1s[15],xc2s[15],ycs[15];
   char  x1s[15],y1s[15],x2s[15],y2s[15];
   char  string[15];
   char  string1[15];
   char  string2[15];
   char  extension[] = ".sr";
   char  extension1[] =".hp";
   char  name[] = "loop";
   char  str[]="\033\056Y\033\056\100\073\063\072\033\056I20\073\07317\072\033\
   char  str1[]="\033\056Z\n";

    cond=debug;
    if(cond==1)
    xoff=1.0;
    else if (cond==0)
    xoff=10.0;

   for ( j=1; j<=Numpages;j++)
   {
     k=1; l=0;
     itoa(j, string1,10);
     strcat (string1,extension1);
     fhpgl= fopen( string1,"w");
     fprintf(fhpgl,"%s",str);
     fprintf(fhpgl,"IN;IP;IW;LT;PU;PA0,0;SP1;VS20;\n");
     fprintf(fhpgl,"SP1;\n");

     for ( m=j; m<=num_file ; m+=Numpages)
     {
        l++;
        if (l== (column+1))
        {
          l=1;
          k++;

        }
        itoa(m, string,10);
        strcat (string,extension );
```

```
if ((fline1 = fopen(string,"r"))==NULL)
{
  printf("\n cannot open the file %s",string);
  break;
}
else

rewind(fline1);
while(fscanf(fline1,"%s%s%s%s", temp1,temp2,temp3,temp4)!=EOF)
{
  count_line++;

  if (strcmp(temp1,name)==0)
  {

    count_line=0;
  }
  else
  {
    x1=atof(temp1);
    y1=atof(temp2);
    x2=atof(temp3);
    y2=atof(temp4);

    x1i=(int)((k-1)*(x_size_with_hole+x_offset)*40.0+(x1 + xoff+x_shift
    x2i=(int)((k-1)*(x_size_with_hole+x_offset)*40.0+(x2 + xoff+x_shift

   if(mirror==1)
   {
     if(mirror_page=='f')        /*   mirrors first page */
     {
        hole_page=1;
        if(j==1)
        {
          x1i=(int)((paper_size_width-2*paper_side)*40.0-x1i);
          x2i=(int)((paper_size_width-2*paper_side)*40.0-x2i);
        }
     }
     else if(mirror_page=='l')     /*   mirrors last page  */
     {
        hole_page=Numpages;
        if(j==Numpages)
        {
          x1i=(int)((paper_size_width-2*paper_side)*40.0-x1i);
          x2i=(int)((paper_size_width-2*paper_side)*40.0-x2i);
        }
     }
   }
   else
   hole_page=Numpages;

    y1i=(int)((l-1)*(y_size+y_offset)*40.0+(y1 + y_shift)*40.0 );
    y2i=(int)((l-1)*(y_size+y_offset)*40.0+(y2 + y_shift)*40.0 );


    itoa(x1i,x1s,10);
    itoa(x2i,x2s,10);
    itoa(y1i,y1s,10);
    itoa(y2i,y2s,10);
```

```
        if(count_line==1)
        {

        fprintf(fhpgl,"PU;PA%s,%s;\n",x1s,y1s);
        fprintf(fhpgl,"PD;PA%s,%s;\n",x2s,y2s);
        }
        else
        fprintf(fhpgl,"PA%s,%s;\n",x2s,y2s);
      }
    }
    rewind(fline1);
    fclose(fline1);
}
 fprintf(fhpgl,"PU;PA1200,6940CI127;\n");
 fprintf(fhpgl,"PA1200,6940CI127;\n");
 fprintf(fhpgl,"PA1200,6940CI127;\n");
 fprintf(fhpgl,"PU,PA9200,6940CI127;\n");
 fprintf(fhpgl,"PA9200,6940CI127;\n");
 fprintf(fhpgl,"PA9200,6940CI127;\n");
 fprintf(fhpgl,"PU;PU;PA0,0;");
 fprintf(fhpgl,"%s", str1);

 /*  The following code generates the registration holes
   for the last page.  The hole diameter is 1/4 inch */

     if (j==hole_page)
     {
       for(k=1;k<=row;k++)
       {
         for(l=1;l<=column;l++)
         {
           xc=((k-1)*(x_size_with_hole+x_offset)*40.0+(xoff/2.0+x_shift)*40
           xc1=(int)((k-1)*(x_size_with_hole+x_offset)*40.0+(xoff/2.0+x_shi
           xc2=(int)(xc + x_size_with_hole*40.0 );

           if(mirror==1)
           {
             xc1=(int)((paper_size_width-2*paper_side)*40.0-xc1);
             xc2=(int)((paper_size_width-2*paper_side)*40.0-xc2);
           }
           yc =(int)((l-1)*(y_size+y_offset)*40.0+(y_size/2.0+y_shift)*40.0
           itoa(xc1,xc1s,10);
           itoa(xc2,xc2s,10);
           itoa(yc,ycs,10);
           fprintf(fhpgl,"PU;PA%s,%sCI127;\n",xc1s,ycs);
           fprintf(fhpgl,"PA%s,%sCI127;\n",xc1s,ycs);
           fprintf(fhpgl,"PA%s,%sCI127;\n",xc1s,ycs);
           fprintf(fhpgl,"PU;PA%s,%sCI127;\n",xc2s,ycs);
           fprintf(fhpgl,"PA%s,%sCI127;\n",xc2s,ycs);
           fprintf(fhpgl,"PA%s,%sCI127;\n",xc2s,ycs);
         }
       }
     }

 fprintf(fhpgl,"PU;PU;PA0,0;");
 fprintf(fhpgl,"%s", str1);
 fclose(fhpgl);
 clrscr();
```

```
        printf("\Create %d HPGL file\n",j);
      }
    return ;
}
```

```
/*****************************************************************************
  PLOT1A.CPP
  Shapemaker 1 Library function
  This program will open an .hp file and output HPGL code to the plotter,
  using the hardware handshake between the computer and the plotter.
  Created by Zetian Wang
  Feb 4 1994
*****************************************************************************/

#include "menu.h"


void plot_out(int num)

{

  char  extension[] =".hp";
  char  string[25];
  char  tmpstring[80],tmpstring1[80];

          itoa(num, string,10);
          strcat (string,extension );

          sprintf(tmpstring,"MODE COM2:9600,N,8,1,p >temp" );
          system(tmpstring);
          sprintf(tmpstring1,"COPY %s COM2: >temp",string );
          system("del temp");
          system(tmpstring1);


          return ;
}
```

CLAIMS

What is claimed is:

1. In a rapid prototype system wherein an object is decomposed into a series of layers, and a physical model of that object is then constructed by creating a first such layer and thereafter sequentially creating additional such layers and bonding each such additional layer to a previous layer, the improvement which comprises:

electronically decomposing said object into thick layers selected such that said thick layers may be positioned across an area corresponding to a sheet of construction material;

electronically slicing said layers into cross sections the thickness of said sheet of construction material;

plotting physical slices corresponding to said cross sections;

forming said physical slices from said construction material;

stacking said physical slices to construct said layers; and

stacking said layers to recompose a physical model of said object.

2. An improvement according to Claim 1, further including the step of selecting said thick slices to minimize the number of layers required to recompose said physical model.

3. An improvement according to Claim 1 wherein said layers are defined by parting planes selected at regions of said object having relatively large dimensions, whereby to minimize overhangs during the step in which said physical slices are stacked to construct said layers.

4. An improvement according to Claim 3 wherein a said layer is constructed as a mirror image, whereby to increase the area of contact between said layer and a support structure upon which said layer is constructed.

5. A rapid prototype method, comprising:

a. electronically decomposing an object into a series of relatively thick layers;

b. electronically slicing said thick layers into cross sections the thickness of a sheet of construction material;

c. plotting on sheets of said construction material physical slices corresponding to said cross sections;

d. cutting said physical slices from said sheets of construction material;

e. stacking said physical slices to construct said layers; and

f. stacking said layers to recompose a physical model of said object.

6. A method according to Claim 5, wherein said sheets of construction material include a construction layer and a backing layer fixed to said construction layer with adhesive material and wherein said physical slices are cut from said construction layer, leaving said backing layer intact.

7. A method according to Claim 6, wherein:

individual pluralities of said physical slices are distributed in corresponding patterns among an ordered set of said construction sheets;

step c includes locating index positions on said sheets of construction material; and

step d includes the placement of first registration holes at selected said index positions through said construction sheets;

whereby to facilitate the precise registration of respective said pluralities of said physical slices carried by individual said construction sheets within said set of construction sheets when all of said construction sheets are stacked in the order of said set with registration pins inserted through said registration holes.

8. A method according to Claim 7, wherein second registration holes are located at second selected index locations to facilitate the precise registration of said layers when they are stacked to recompose said object.

9. A method according to Claim 5, wherein steps a, b, c and d are controlled by software equivalent to that of the appendix.

10. A method according to Claim 5, wherein:

said construction material is in the form of flat sheets, each having a construction layer bonded to a backing layer; and

step d is conducted such that said slices are cut through said construction layer, leaving said backing layer in tact.

11. A method according to Claim 10, wherein steps c and d are conducted such that registration holes are cut through said backing layer so that corresponding slices carried by respective said backing layers may be precisely registered during step e.

12. A method according to Claim 11, wherein steps a, b, c and d are controlled by software equivalent to that of the appendix.

13. A method according to Claim 7, wherein steps e and f are conducted on a build platform including first registration pins positioned to receive said first registration holes provided in said sheets.

14. A method according to Claim 13, wherein said build platform includes second registration pins positioned to receive second registration holes provided in said sheets, said second registration holes being located to facilitate the precise registration of said layers when they are stacked to recompose said object during step f.

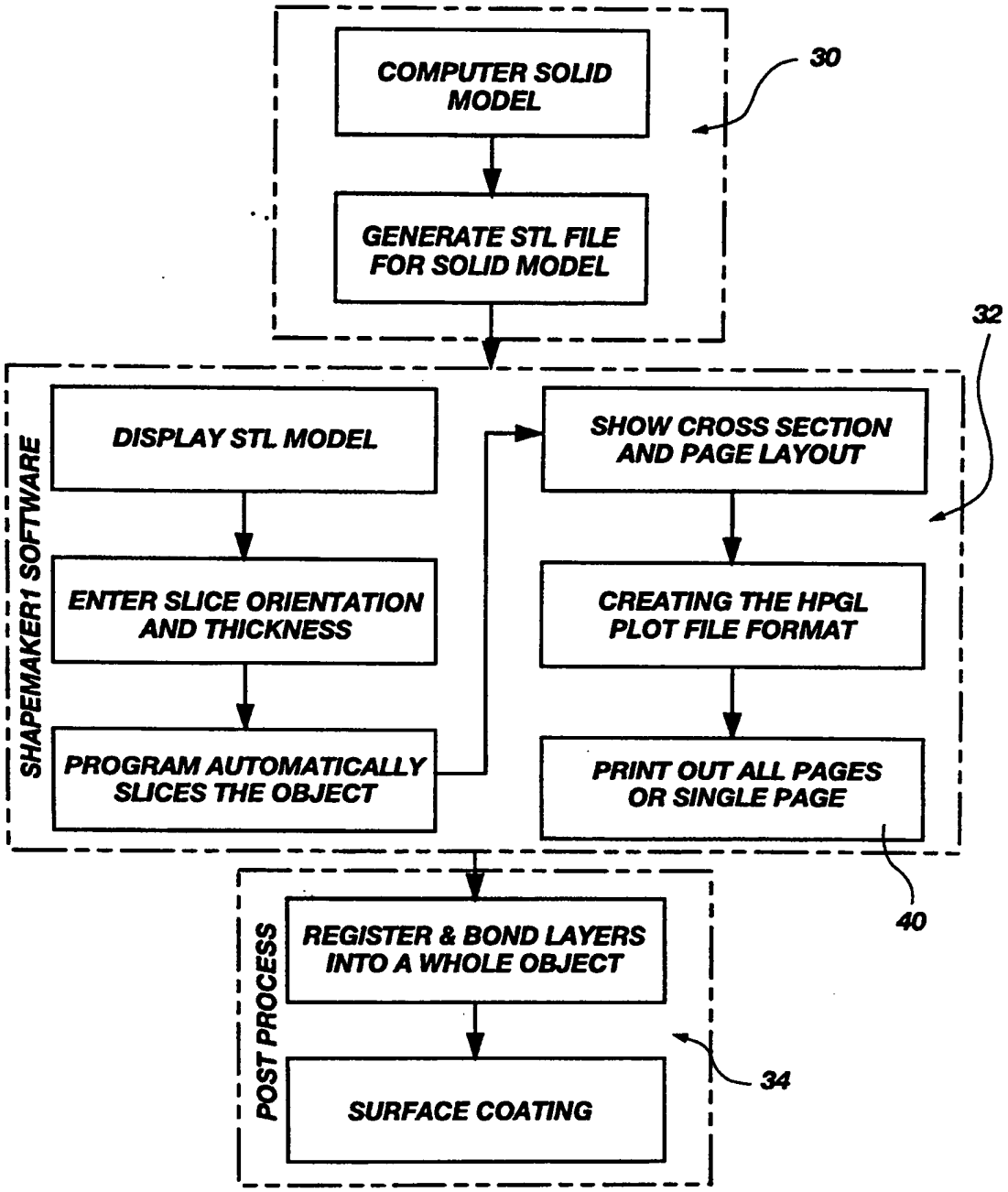

COMPUTER SOLID MODEL
30
GENERATE STL FILE FOR SOLID MODEL
32
SHAPEMAKER1 SOFTWARE
DISPLAY STL MODEL
ENTER SLICE ORIENTATION AND THICKNESS
PROGRAM AUTOMATICALLY SLICES THE OBJECT
SHOW CROSS SECTION AND PAGE LAYOUT
CREATING THE HPGL PLOT FILE FORMAT
PRINT OUT ALL PAGES OR SINGLE PAGE
40
POST PROCESS
REGISTER & BOND LAYERS INTO A WHOLE OBJECT
SURFACE COATING
34

Fig. 1

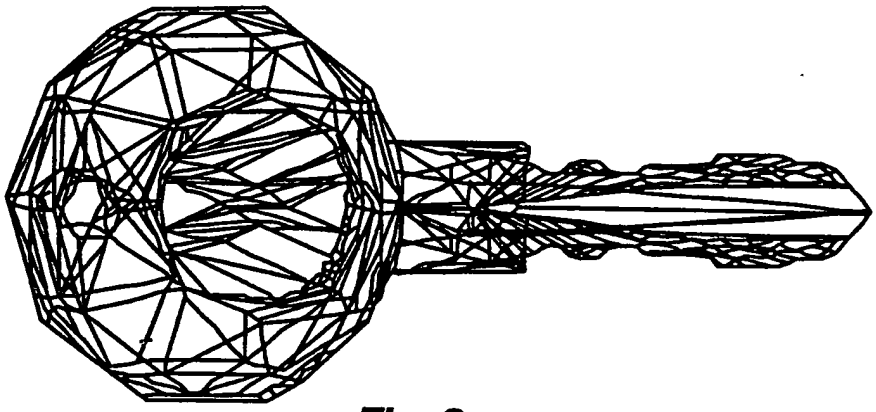


Fig. 2a

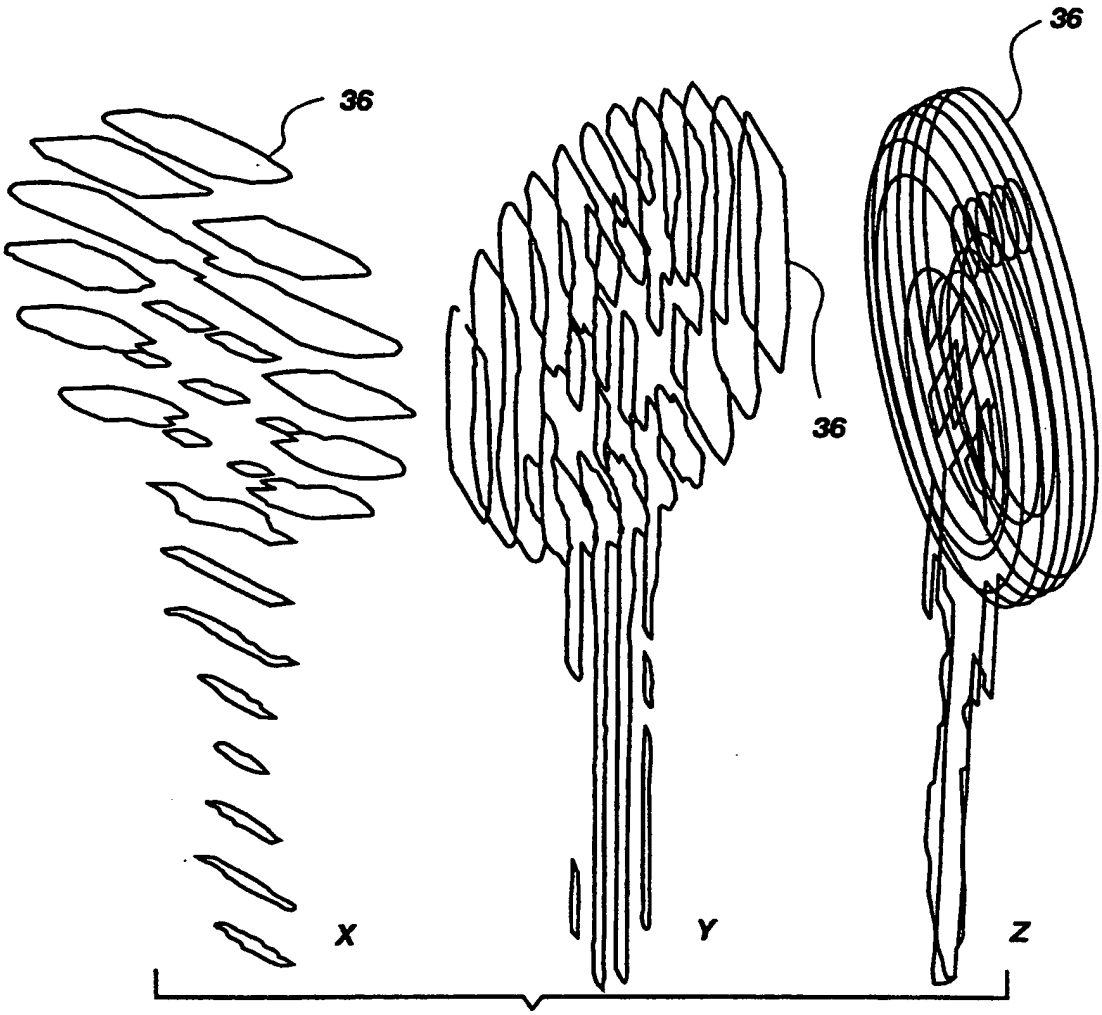


Fig. 2b

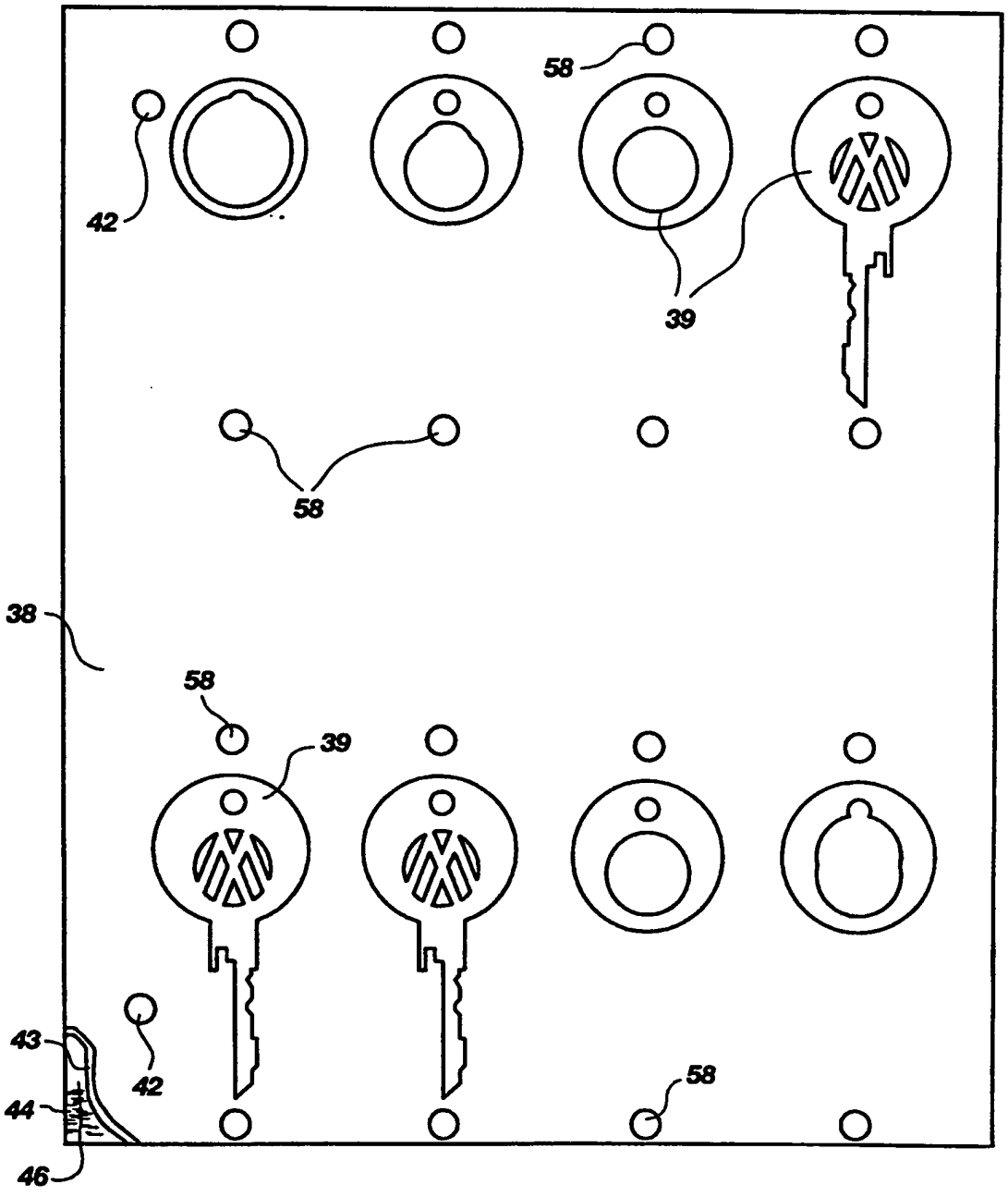


Fig. 3

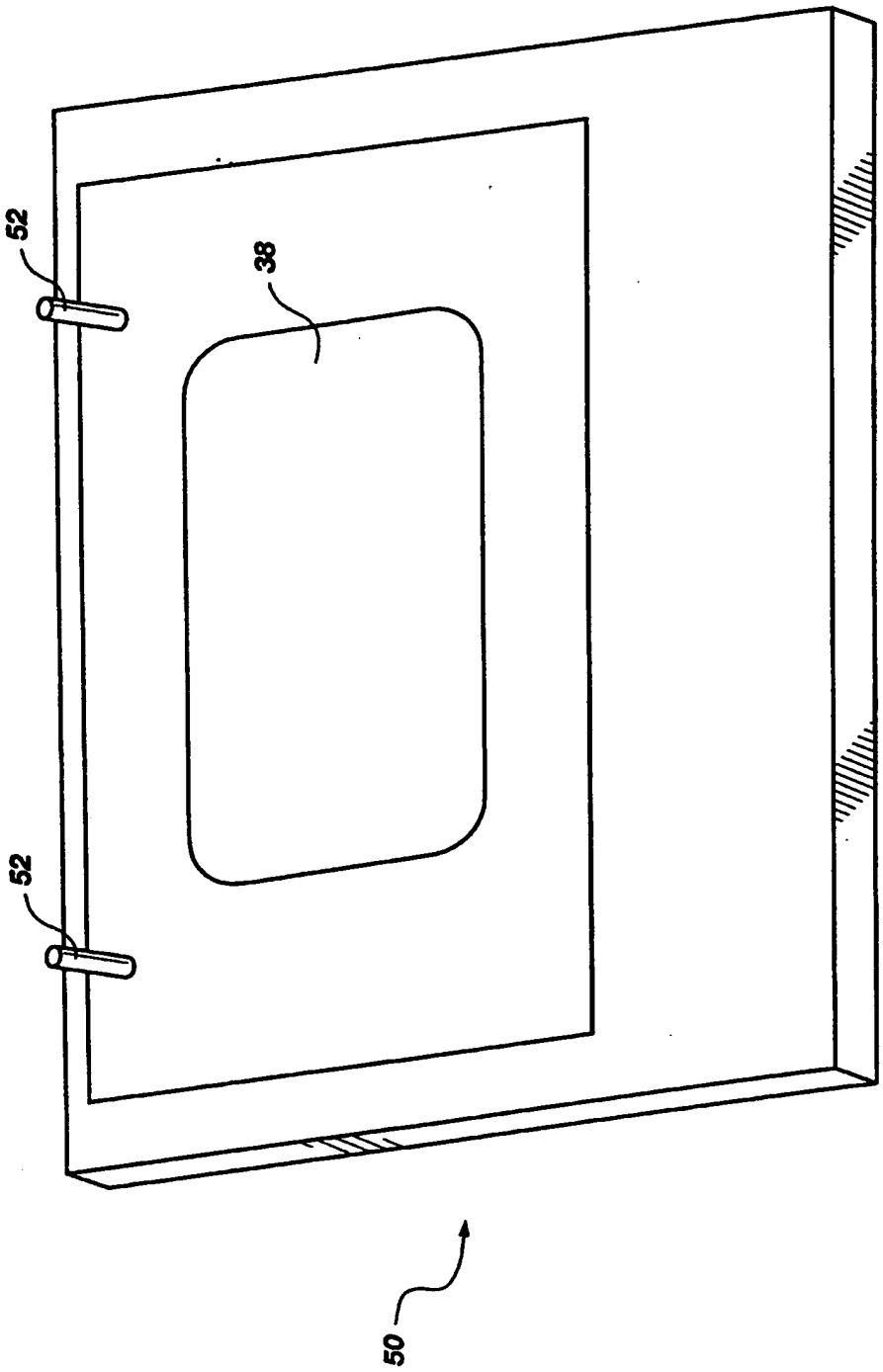


Fig. 4

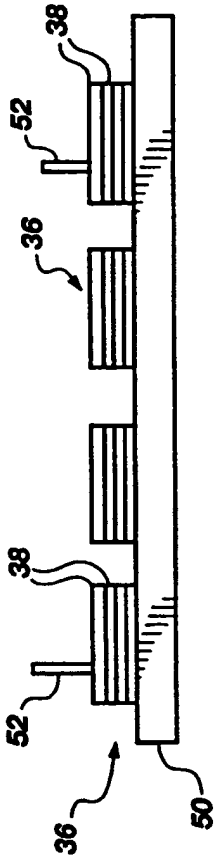


Fig. 5

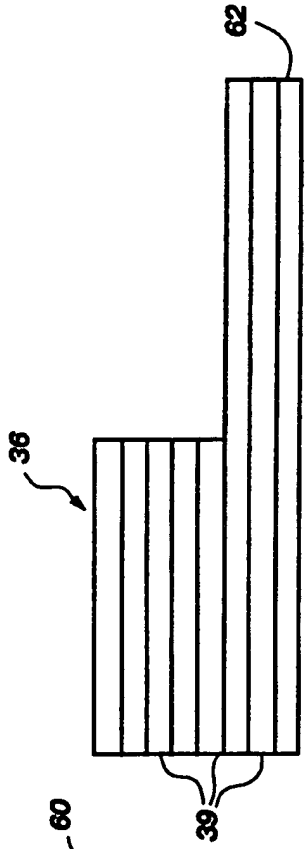


Fig. 6b

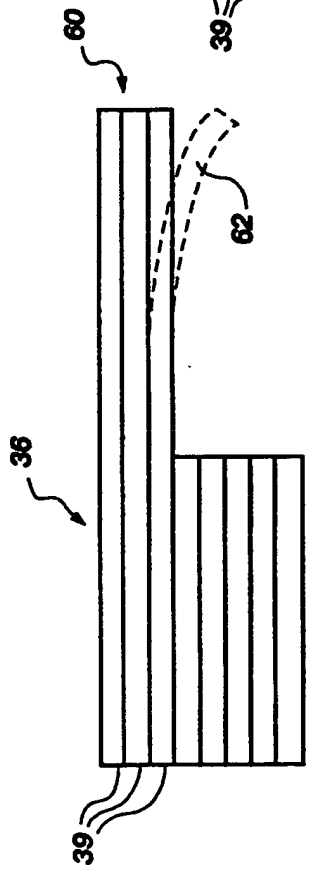


Fig. 6a

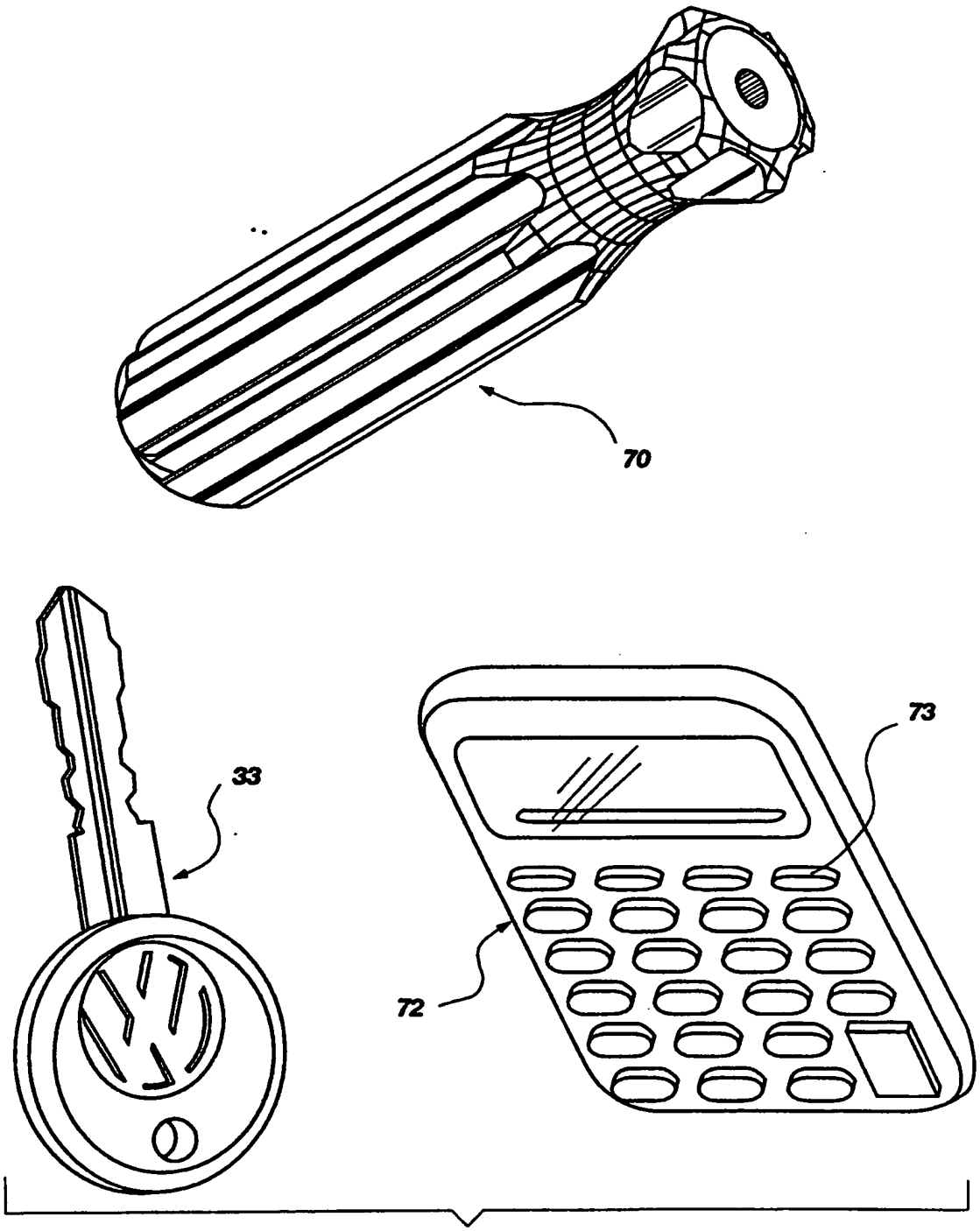


Fig. 7

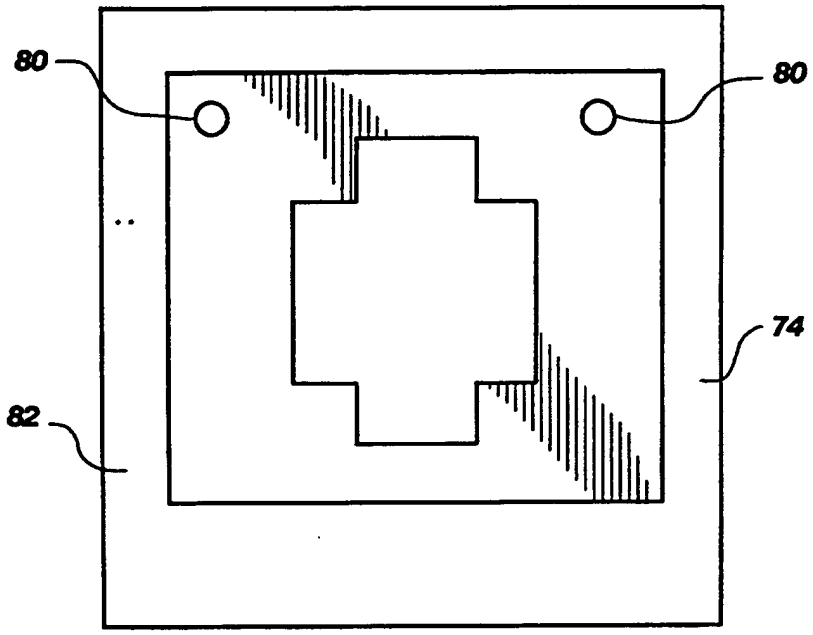


Fig. 8a

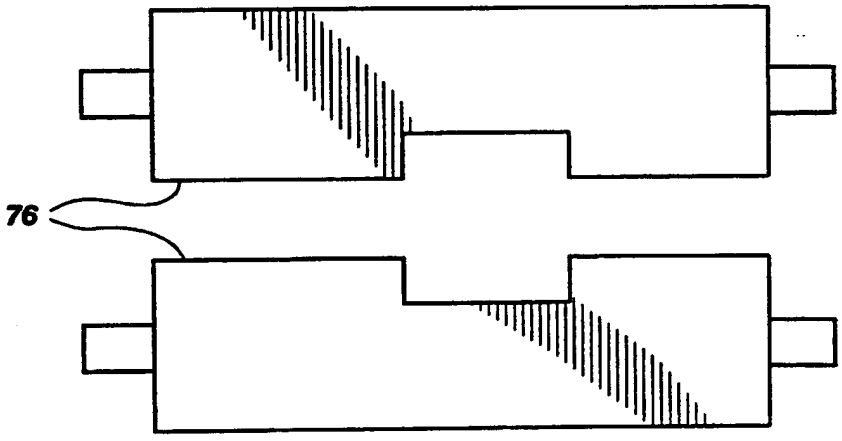


Fig. 8b

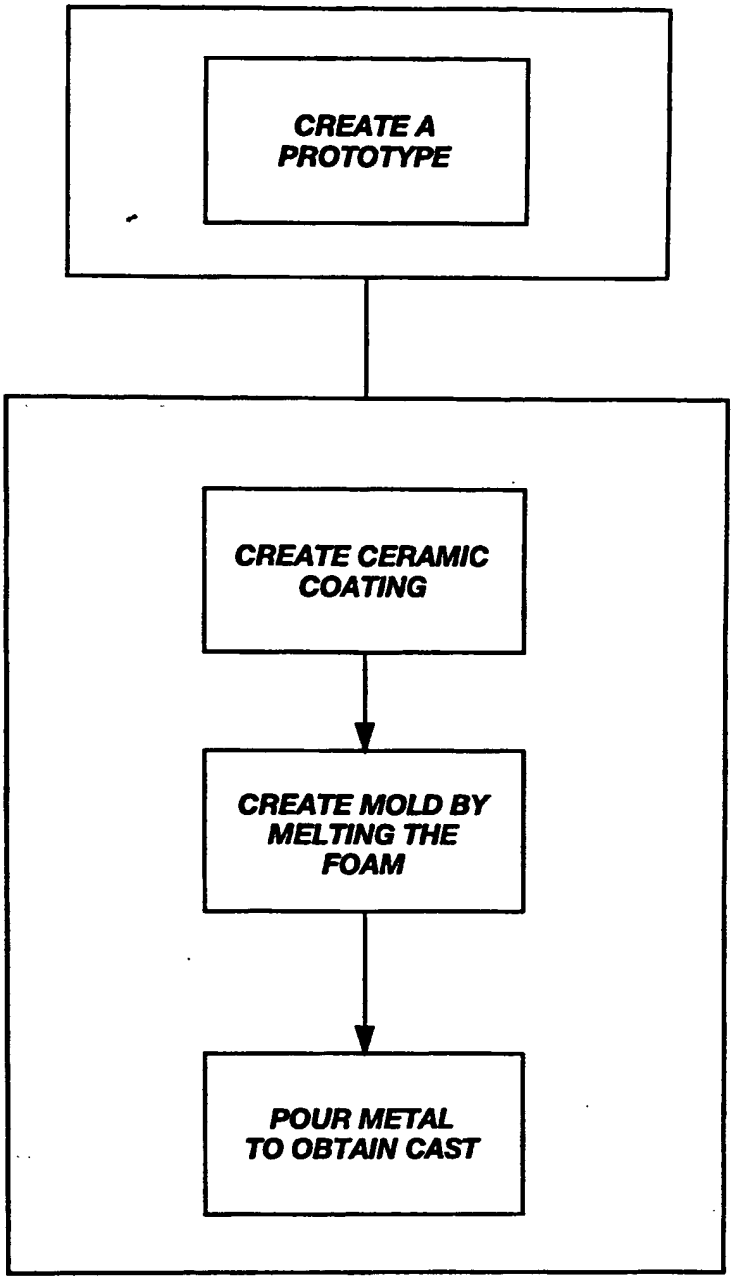

CREATE A PROTOTYPE
CREATE CERAMIC COATING
CREATE MOLD BY MELTING THE FOAM
POUR METAL TO OBTAIN CAST

Fig. 9

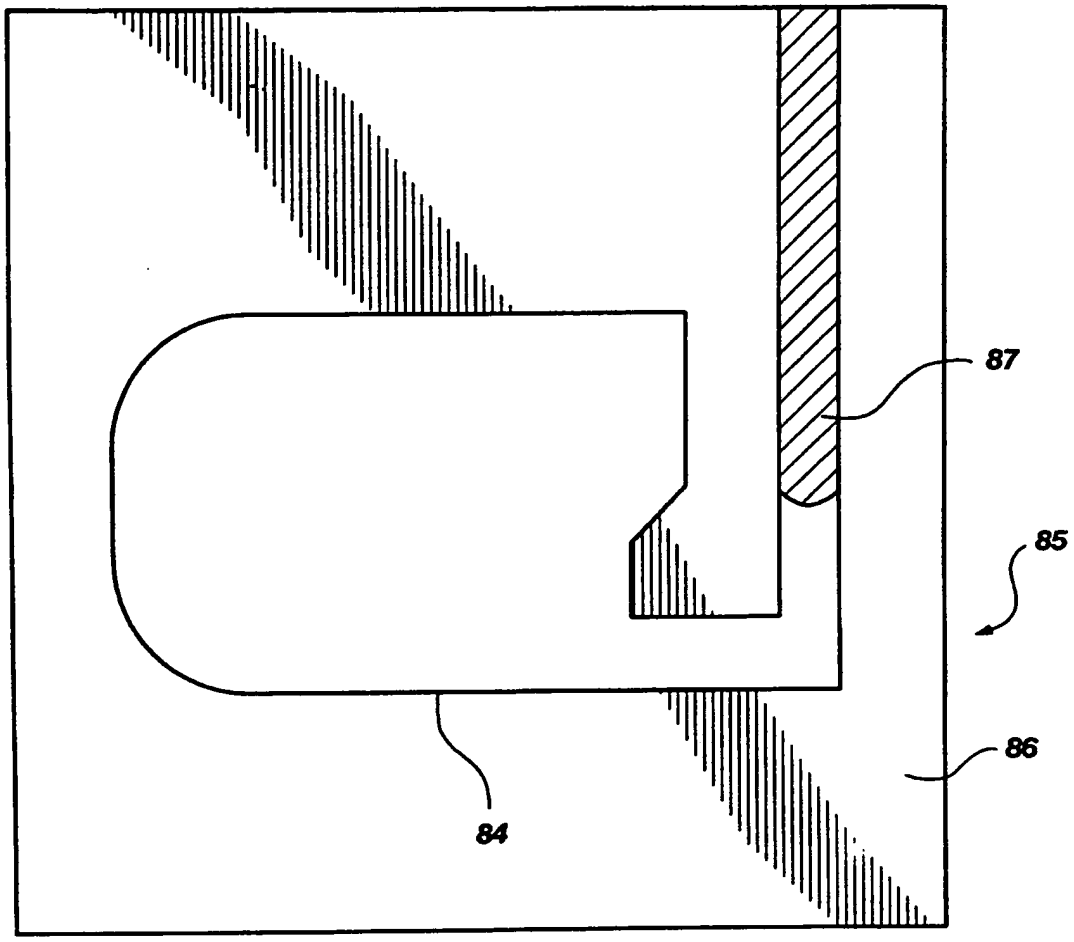


Fig. 10

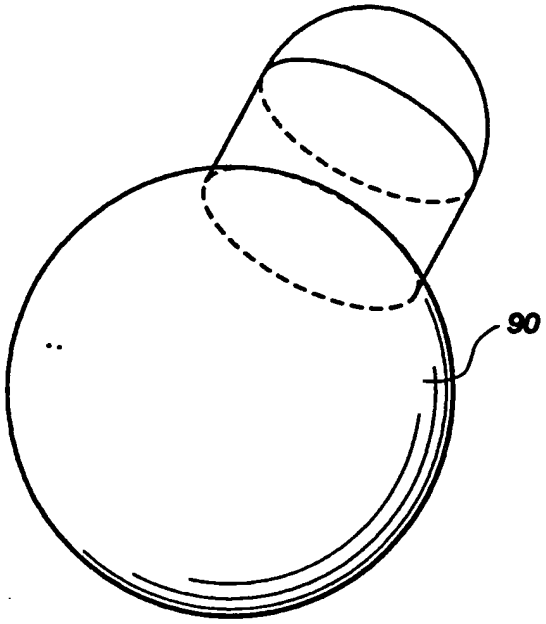


Fig. 11a

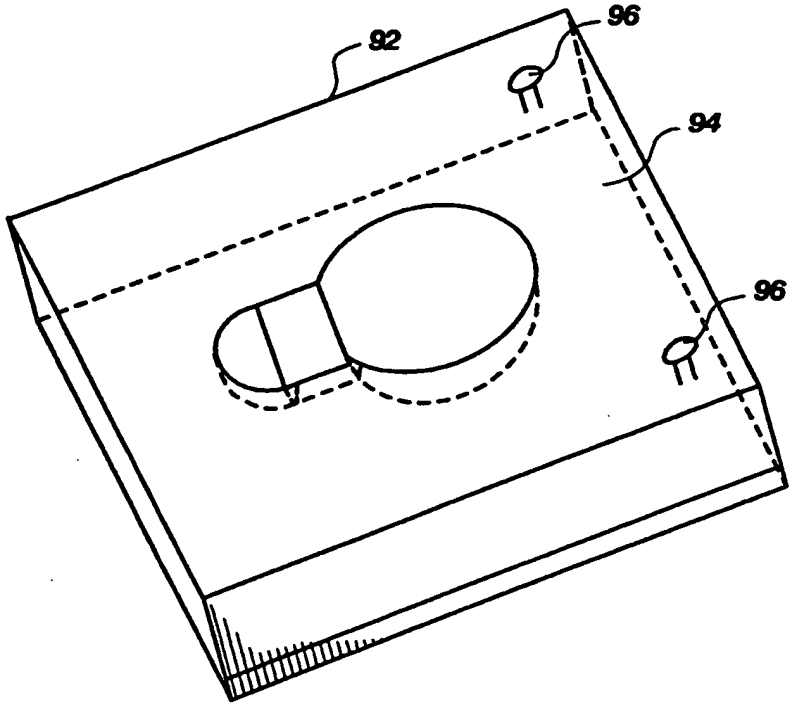


Fig. 11b

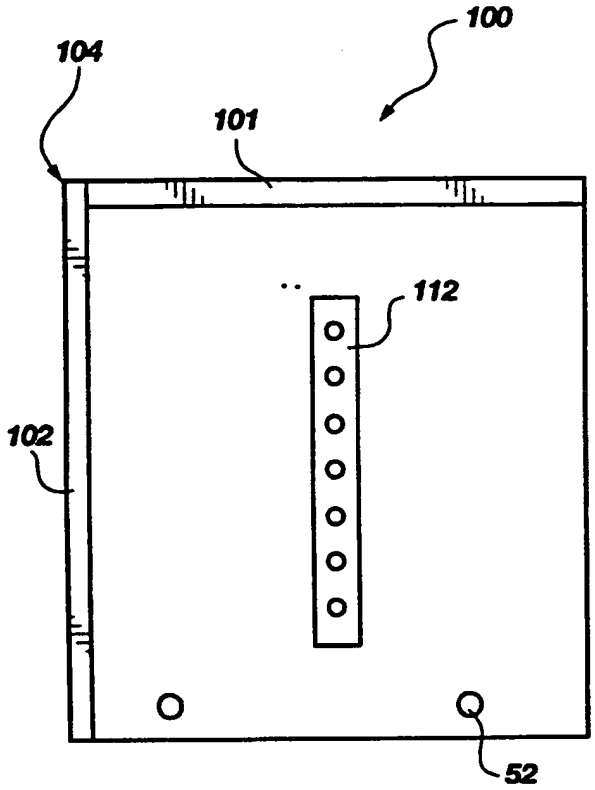


Fig. 12a

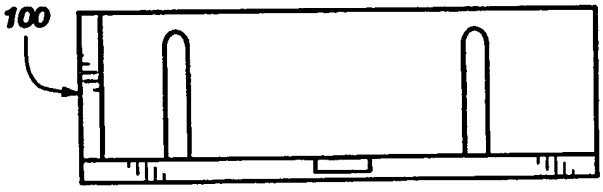


Fig. 12b

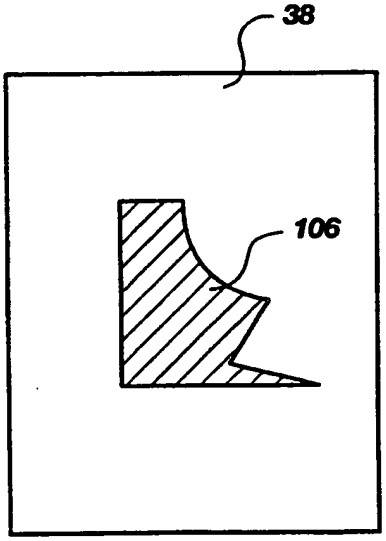


Fig. 12c

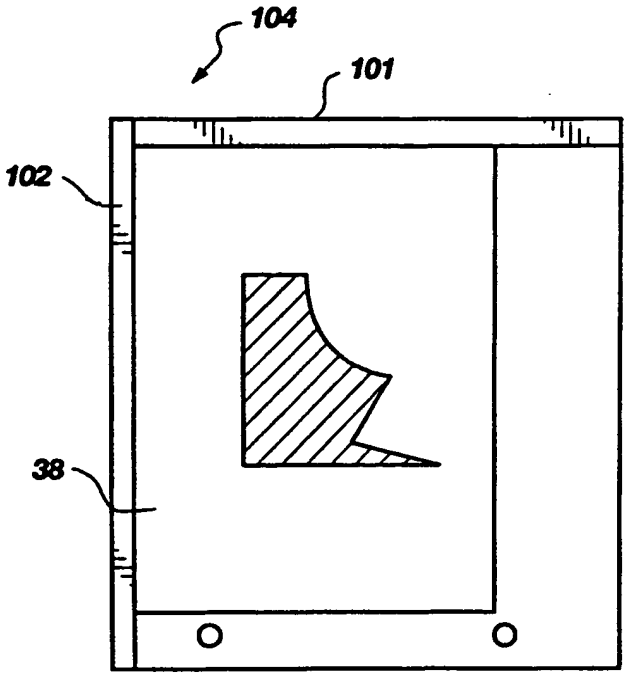


Fig. 13a

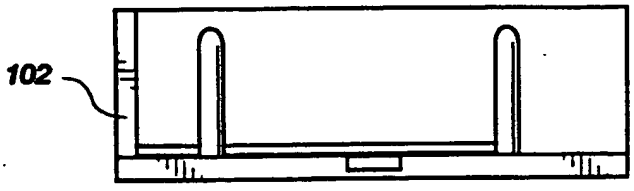


Fig. 13b

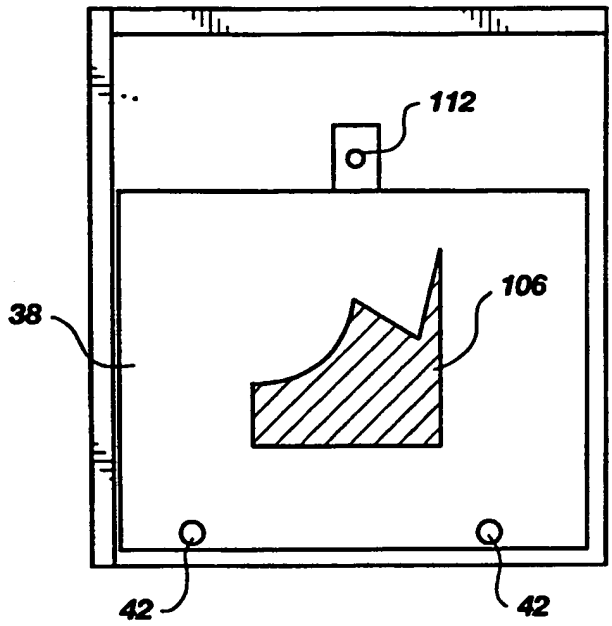


Fig. 14a

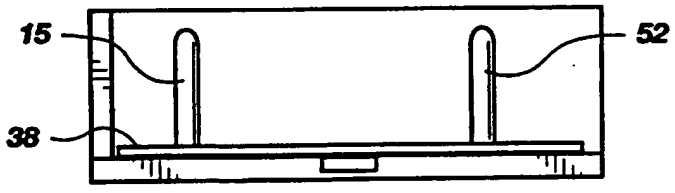


Fig. 14b

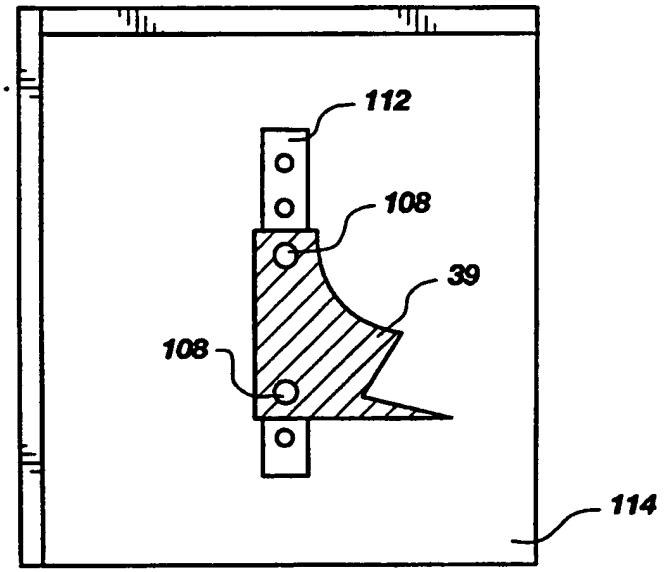


Fig. 15a

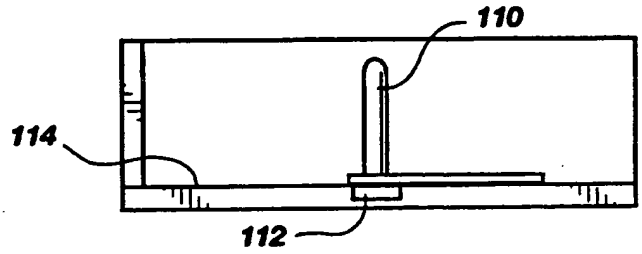


Fig. 15b

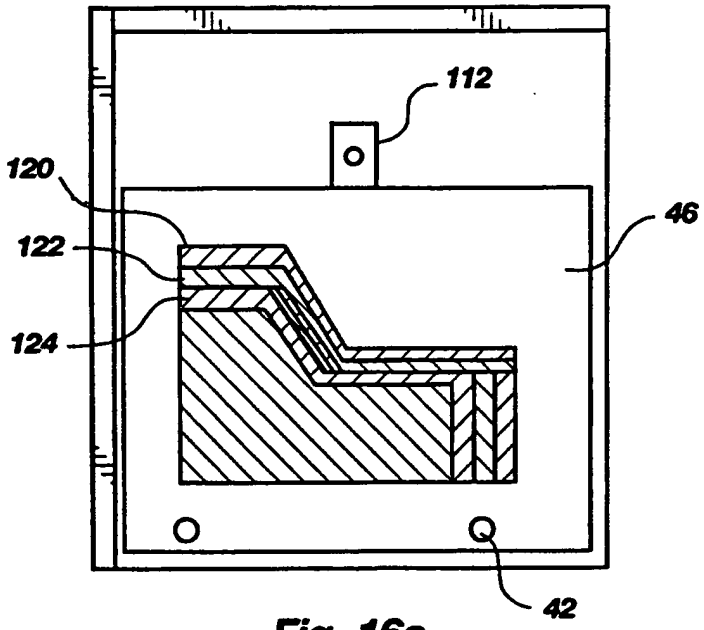


Fig. 16a

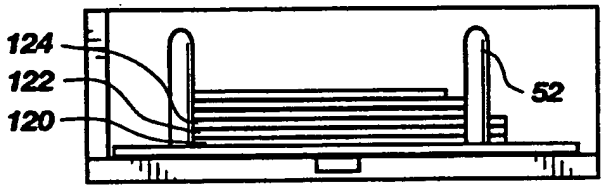


Fig. 16b

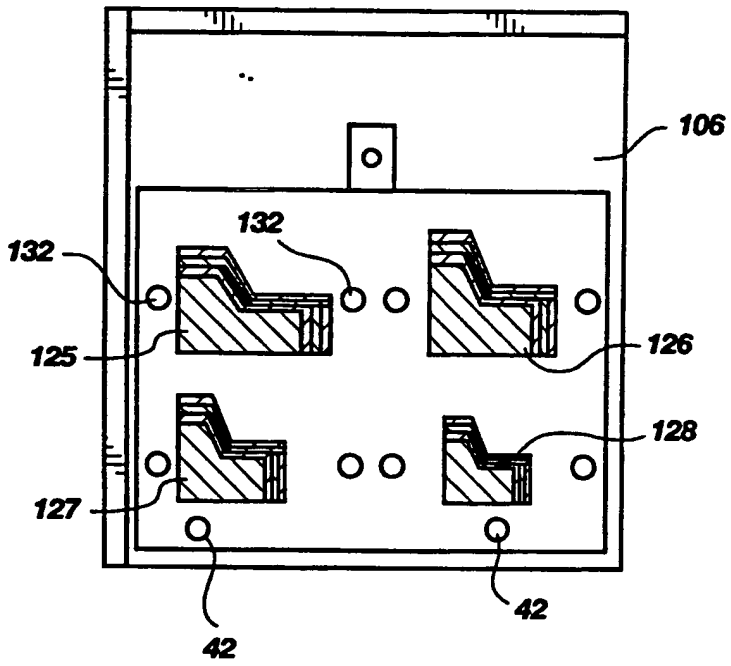


Fig. 17a

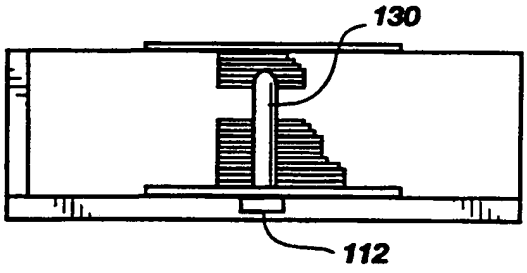


Fig. 17b

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US96/13486

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :Please See Extra Sheet.
US CL :364/468.26,474.24

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 364/468.26-468.27,468.24,468.25,474.24; 395/118-120;156/59,250,256,263-265,379.8,517

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,P	US, A, 5,514,232 (BURNS) 07 MAY 1996 (07.05.96) column 3, lines 13-43, figure 1.	1-14
A	US,A, 5,088,047 (BYNUM) 11 FEBRUARY 1992 (11.02.92) column 3, lines 36-62, figure 1.	1-14

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
04 NOVEMBER 1996	29 NOV 1996
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	Authorized officer STEVEN R. GARLAND
Facsimile No. (703) 305-3230	Telephone No. (703) 305-3800

Form PCT/ISA/210 (second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US96/13486

A. CLASSIFICATION OF SUBJECT MATTER:
IPC (6):

G06F 19/00

PCT ORGANISATION MONDIALE DE LA PROPRIETE INTELLECTUELLE
Bureau international

DEMANDE INTERNATIONALE PUBLIEE EN VERTU DU TRAITE DE COOPERATION EN MATIERE DE BREVETS (PCT)

(51) Classification internationale des brevets 7 : **G05B 19/4099**	**A1**	**(11) Numéro de publication internationale:** **WO 00/31600** **(43) Date de publication internationale:** 2 juin 2000 (02.06.00)

(21) Numéro de la demande internationale: PCT/FR99/02790

(22) Date de dépôt international: 15 novembre 1999 (15.11.99)

(30) Données relatives à la priorité:
98/14687 19 novembre 1998 (19.11.98) FR

(71) Déposant *(pour tous les Etats désignés sauf US)*: C.I.R.T.E.S. (CENTRE D'INGENIERIE DE RECHERCHE ET DE TRANSFERT DE L'ESSTIN A SAINT-DIE) [FR/FR]; 29 bis, rue d'Hellieule, F-88100 Saint-Dié (FR).

(72) Inventeur; et
(75) Inventeur/Déposant *(US seulement)*: BARLIER, Claude [FR/FR]; 67, chemin de la Roche, F-88100 Coinches (FR).

(74) Mandataire: POUPON, Michel; Cabinet Michel Poupon, 3, rue Ferdinand Brunot, F-88026 Epinal Cedex (FR).

(81) Etats désignés: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZA, ZW, brevet ARIPO (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), brevet eurasien (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), brevet européen (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), brevet OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

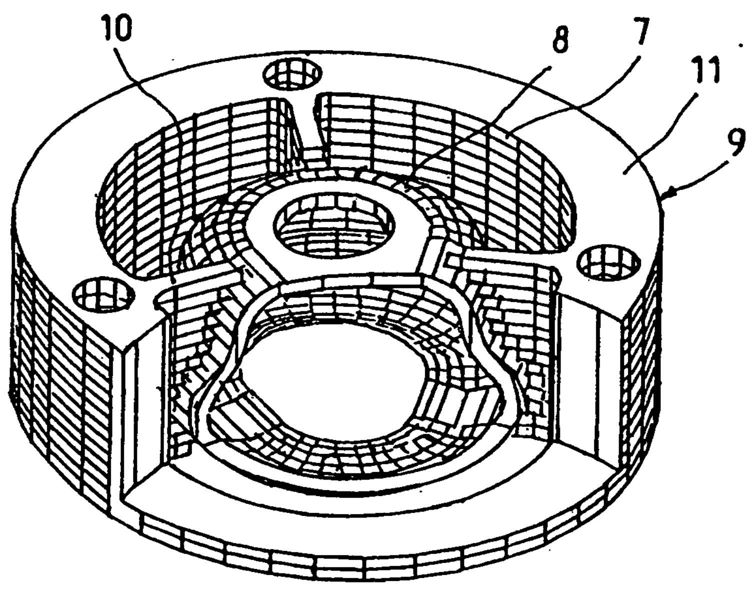
Publiée
Avec rapport de recherche internationale.

(54) Title: METHOD FOR MAKING MECHANICAL PARTS BY DECOMPOSITION INTO LAYERS

(54) Titre: PROCEDE DE REALISATION DE PIECES MECANIQUES PAR DECOMPOSITION EN STRATES

(57) Abstract

The invention concerns a method for making mechanical parts and objects, in particular prototypes, from a specific computer-assisted design comprising the following successive steps: making the parts in elementary layers or strata; reconstructing the assembly of layers; assembling the layers, said layers being derived from previous decomposition according to specific planes and one or several step(s). The invention is characterised in that the unit layers determined by the decomposition of the part using a software and machined accordingly comprise essentially: a central portion (8) effectively corresponding to the layer having the shape and thickness desired for the finished part; an outer portion (11) substantially of same thickness, enclosing at least partly said central portion; cleavable hasps (10) linking said central and outer portions together.



(57) Abrégé

Procédé de réalisation de pièces mécaniques et objets, en particulier de prototypes, à partir d'une conception assistée par ordinateur spécifique du type comportant les phases successives de: fabrication des pièces en couches ou strates élémentaires; reconstitution de l'ensemble des couches; assemblage des couches; lesdites strates étant issues d'une décomposition préalable de la pièce selon des plans et un ou des pas déterminés, caractérisé en ce que les strates unitaires déterminées par la décomposition de la pièce mettant en oeuvre un logiciel et usinées en conséquence comportent essentiellement: une partie centrale (8) correspondant effectivement à la strate ayant la forme et l'épaisseur recherchées pour l'obtention de la pièce finie; une partie extérieure (11) sensiblement de même épaisseur, enveloppant au moins partiellement ladite partie centrale; des pontets sécables (10) reliant lesdites parties centrale et extérieure entre elles.

UNIQUEMENT A TITRE D'INFORMATION

Codes utilisés pour identifier les Etats parties au PCT, sur les pages de couverture des brochures publiant des demandes internationales en vertu du PCT.

AL	Albanie	**ES**	Espagne	**LS**	Lesotho	**SI**	Slovénie
AM	Arménie	**FI**	Finlande	**LT**	Lituanie	**SK**	Slovaquie
AT	Autriche	**FR**	France	**LU**	Luxembourg	**SN**	Sénégal
AU	Australie	**GA**	Gabon	**LV**	Lettonie	**SZ**	Swaziland
AZ	Azerbaïdjan	**GB**	Royaume-Uni	**MC**	Monaco	**TD**	Tchad
BA	Bosnie-Herzégovine	**GE**	Géorgie	**MD**	République de Moldova	**TG**	Togo
BB	Barbade	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tadjikistan
BE	Belgique	**GN**	Guinée	**MK**	Ex-République yougoslave de Macédoine	**TM**	Turkménistan
BF	Burkina Faso	**GR**	Grèce			**TR**	Turquie
BG	Bulgarie	**HU**	Hongrie	**ML**	Mali	**TT**	Trinité-et-Tobago
BJ	Bénin	**IE**	Irlande	**MN**	Mongolie	**UA**	Ukraine
BR	Brésil	**IL**	Israël	**MR**	Mauritanie	**UG**	Ouganda
BY	Bélarus	**IS**	Islande	**MW**	Malawi	**US**	Etats-Unis d'Amérique
CA	Canada	**IT**	Italie	**MX**	Mexique	**UZ**	Ouzbékistan
CF	République centrafricaine	**JP**	Japon	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Pays-Bas	**YU**	Yougoslavie
CH	Suisse	**KG**	Kirghizistan	**NO**	Norvège	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	République populaire démocratique de Corée	**NZ**	Nouvelle-Zélande		
CM	Cameroun			**PL**	Pologne		
CN	Chine	**KR**	République de Corée	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Roumanie		
CZ	République tchèque	**LC**	Sainte-Lucie	**RU**	Fédération de Russie		
DE	Allemagne	**LI**	Liechtenstein	**SD**	Soudan		
DK	Danemark	**LK**	Sri Lanka	**SE**	Suède		
EE	Estonie	**LR**	Libéria	**SG**	Singapour		

Procédé de réalisation de pièces mécaniques par décomposition en strates

La présente invention a pour objet un perfectionnement aux procédés de réalisation de pièc s mécaniques et objets en particulier de prototyp s à partir d'une conception assistée par ordinateur spécifique du type comportant les phases successives de :

- fabrication des pièces en couches ou strates élémentaires ;
- reconstitution de l'ensemble des couches ;
- assemblage des couches ;

lesdites strates étant issues d'une décomposition préalable de la pièce selon des plans et un ou des pas déterminés.

L'invention a également pour objet les strates élémentaires ainsi réalisées, de même que les prototypes obtenus pour l'assemblage desdites strates élémentaires.

Un procédé de prototypage rapide de ce type a fait l'objet du brevet européen EP-0 585 502-B1 dont le contenu est intégré ici entièrement par voie de référence et est connu sous le nom de STRATOCONCEPTION (marque déposée).

Ce procédé donne entière satisfaction dans les limites des applications spécifiées dans ce brevet, le positionnement et l'assemblage des différentes strates étant essentiellement obtenus par des inserts dont la forme et le positionnement sont déterminés également par un logiciel spécifique.

Le fait de prévoir des inserts à l'intérieur, pour des pièces d'une certaine épaisseur, apporte néanmoins une certaine lourdeur au procédé de mise en œuvre, par ailleurs très souple et très performant.

D'autre part, il n'est pas possible de prévoir facilement des inserts à l'intérieur pour des strates dont la section utile (épaisseur de la pièce finale) est faible, strates nécessaires pour l'obtention d'une modélisation très fine, donc plus précise, ou pour la réalisation de pièces dont la structure complexe implique une décomposition passant par des strates de très faible épaisseur latérale.

L'invention a pour objet de proposer un procédé selon le concept général du brevet 0 585 502 permettant en outre de s'abstenir éventuellement de l'utilisation d'inserts d'assemblage à l'intérieur des strates entre elles et de positionnement des strates, l'une par rapport à l'autre.

Conformément à l'invention, ce r´sultat est obtenu avec un procédé de réalisation des pièces mécaniques et objets, en particulier de prototypes, à partir d'une conception assistée par ordinateur spécifique du type comportant les phases successives de :

- fabrication des pièces en couches ou strates élémentaires ;
- reconstitution de l'ensemble des couches ;
- assemblage des couches ;

lesdites strates étant issues d'une décomposition préalable de la pièce selon des plans et un ou des pas déterminés, caractérisé en ce que les strates unitaires déterminées par la décomposition de la pièce mettant en œuvre un logiciel et usinées en conséquence comportent essentiellement :

- une partie centrale correspondant effectivement à la strate ayant la forme et l'épaisseur recherchées pour l'obtention de la pièce finie,
- une partie extérieure sensiblement de même épaisseur, enveloppant au moins partiellement ladite partie centrale,
- des pontets sécables reliant lesdites parties centrale et extérieure entre elles.

L'assemblage des strates entre elles est ensuite opéré par superposition ou

échafaudage des différentes strates, les parties extérieures de chaque strate formant finalement une sorte d'enveloppe porteuse enserrant la pièce reconstituée à laquelle elle est reliée par les pontets sécables.

On comprendra que la décomposition de la pièce et l'assemblage sont obtenus de manière systématique par l'utilisation du logiciel spécifique qui positionne et prévoit automatiquement les pontets, les piliers, les inserts intérieurs ou extérieurs.

Ainsi, des inserts de positionnement et de maintien sont rapportés sur l'enveloppe extérieure. Ceux-ci permettent de positionner les strates de manière indirecte par montage et assemblage (par exemple, mais non limitativement par collage).

L'enveloppe porteuse est ensuite supprimée facilement, du fait des pontets sécables, après positionnement des strates et assemblage.

L'enveloppe englobera la pièce finale au plus près, pour des raisons de précision d'assemblage et d'économie de matière, ce qui nécessite dans tous les cas de figure un système de pressage par bridage.

Le système de pressage pourra être extérieur, par exemple avec une plaque de montage, ou intégré, l'enveloppe étant autoporteuse.

On comprendra mieux l'invention à l'aide de la description ci-après faite en référence aux dessins annexés dans lesquels :

- la figure 1 illustre schématiquement sous forme d'un diagramme le principe de mise en œuvre du procédé dit de stratoconception ;
- la figure 2 illustre schématiquement une pièce reconstituée à partir de strates élémentaires avec enveloppe extérieure, conformément à l'invention ;
- les figures 3A à 3F représentent des variantes de réalisation des pontets sécables et des enveloppes extérieures ;
- la figure 4 représente la pièce de la figure 2 avec une structure de maintien et d'assemblage autoporteuse ;
- la figure 5 représente une variante de la pièce de la figure 2 avec une structure de maintien et d'assemblage faisant appel à une plaque de montage ;
- les figures 6 et 7 représentent une variante de la pièce de la figure 2, avec une variante d'assemblage de même type que celui des figures 2 et 5 ;
- la figure 8 illustre en coupe partielle un assemblage possible avec inserts extérieurs pour des formes complexes et des strates fines.

On se référera tout d'abord à la figure 1.

Le principe général consiste, par la mise en œuvre d'un logiciel spécifique (1), à

découper en strates une pièce à prototyper, les strates étant usinées par micro fraisage rapide (2), la machine étant pilotée par le logiciel (1), d'un matériau en plaque (3).

Les différentes strates sont assemblées selon un ensemble (4) comportant des inserts (5) pour obtenir finalement un prototype (6) après finition.

Le logiciel gère le choix du plan de tranchage/stratification, du pas du profil de strate, du rapport d'échelle, de la précision, du positionnement des inserts.

Après la saisie des différents paramètres de plaque (dimensions ; matériau, choix du sens de dépouille) et des paramètres d'usinage (vitesse de coupe, diamètre de fraise, etc...) l'ensemble du programme d'usinage est transmis par le logiciel qui pilote le robot de découpe.

On se référera maintenant à la figure 2 .

Selon l'invention, le procédé mis en œuvre permet d'obtenir une multitude de strates élémentaires (7) qui, une fois assemblées, reproduisent la pièce à reproduire (8) reliée à une enveloppe extérieure (9) par des pontets (10).

On comprendra que, après assemblage adéquat, l'élimination de l'enveloppe et des pontets aboutisse à l'obtention de la pièce finale (8) en particulier prototype.

Les strates (7) peuvent être de formes géométriques différentes et très variées au niveau des pontets (10) et des éléments (11) formant finalement l'enveloppe extérieure (9).

Différentes variantes sont représentées de manière non limitative aux figures 3A à 3F.

A la figure 3A on a représenté trois variantes de pontets au niveau de la zone de fragilisation (12) où s'effectuera la découpe.

A la figure 3B, on notera que la répartition des pontets, par exemple au nombre de trois, peut être régulière sur le pourtour de la partie centrale (en l'occurrence ici à 120°).

A la figure 3C, la variante consiste à ce que les éléments (11) soient des secteurs arrondis et enveloppants (13).

A la figure 3D, trois secteurs tels que (13) sont réunis pour former une platine unique (14), qui enserre totalement la pièce selon une couronne (15) à la représentation de la figure 3E.

Enfin, à la figure 3F, les orifices (16') présents dans chaque structure et servant à positionner et assembler les pontets entre eux seront non plus circulaires (16) comme aux figures précédentes, mais à section géométrique polygonale, ce qui permet d'en limiter le nombre sur une même strate pour un même positionnement précis.

Les strates sont assemblées sur des axes (17) qui, à la figure 4, sont au nombre de trois, ceux-ci comportant, par exemple mais non limitativement, des écrous papillons (18) de serrage. La structure est ici autoporteuse.

A la figure 5, on prévoit deux axes (21) fixés sur une plaque de montage (19) munie d'alésages (20).

Ce type d'assemblage peut être en outre utilisé lorsque le procédé est mis en œuvre de manière systématique et par exemple à des fins pédagogiques, voire ludiques.

Aux figures 6 et 7, les structures sont identiques à celles des figures 4 et 5, avec un seul axe tel que (17,21) et une tige (22) de type insert pour assurer le positionnement.

Enfin, à la figure 8, on a représenté une variante complexe avec des inserts (23) pour des strates de très faibles épaisseurs, chaque insert concernant uniquement quelques strates jointives.

Bien entendu, chaque strate élémentaire sera usinée par microfraisage conformément au procédé général dit de « Stratoconception », éventuellement avec retournement si nécessaire en cours d'usinage selon le procédé décrit dans une demande de brevet déposée conjointement par la demanderesse et à laquelle il est fait expressément référence.

Ce procédé permet la réalisation de prototypes de pièces de formes très complexes, très rapidement et à faible coût. Il ouvre également des perspectives intéressantes d'applications pédagogiques et ludiques.

REVENDICATIONS

1. Procédé de réalisation de pièces mécaniques et objets, en particulier de prototypes, à partir d'une conception assistée par ordinateur spécifique du type comportant les phases successives de :

- fabrication des pièces en couches ou strates élémentaires ;
- reconstitution de l'ensemble des couches ;
- assemblage des couches ;

lesdites strates étant issues d'une décomposition préalable de la pièce selon des plans et un ou des pas déterminés, caractérisé en ce que les strates unitaires déterminées par la décomposition de la pièce mettant en œuvre un logiciel spécifique et usinées en conséquence comportent essentiellement :

- une partie centrale (8) correspondant effectivement à la strate ayant la forme et l'épaisseur recherchées pour l'obtention de la pièce finie,
- une partie extérieure (11) sensiblement de même épaisseur, enveloppant au moins partiellement ladite partie centrale,
- des pontets sécables (10) reliant lesdites parties centrale et extérieure entre elles.

2. Procédé selon la revendication 1, caractérisé en ce que chaque strate comporte des orifices (16) circulaires de positionnement et d'assemblage des pontets entre eux.

3. Procédé selon la revendication 1, caractérisé en ce que chaque strate comporte des orifices (16') à section géométrique polygonale de positionnement et d'assemblage des pontets entre eux.

4. Procédé selon l'une quelconque des revendications 1 à 3, caractérisé en ce que les strates sont assemblées selon une structure autoporteuse.

5. Procédé selon l'une quelconque des revendications 1 à 3, caractérisé en ce que les strates sont assemblées sur une plaque de montage (19) munie d'alésages (20).

6. Procédé selon l'une quelconque des revendications 1 à 5, caractérisé en ce que l'assemblage est réalisé au moyen d'un axe unique (17,21) et d'une tige insert (22).

7. Strate élémentaire pour la réalisation par assemblage d'une pièce mécanique en particulier prototype, caractérisée en ce qu'elle est obtenue par la mise en œuvre d'un procédé selon l'une quelconque des revendications 1 à 6.

8. Pièce mécanique, en particulier prototype, caractérisée en ce qu'elle est obtenue par l'assemblage de states selon la revendication 7.

FIG. 1

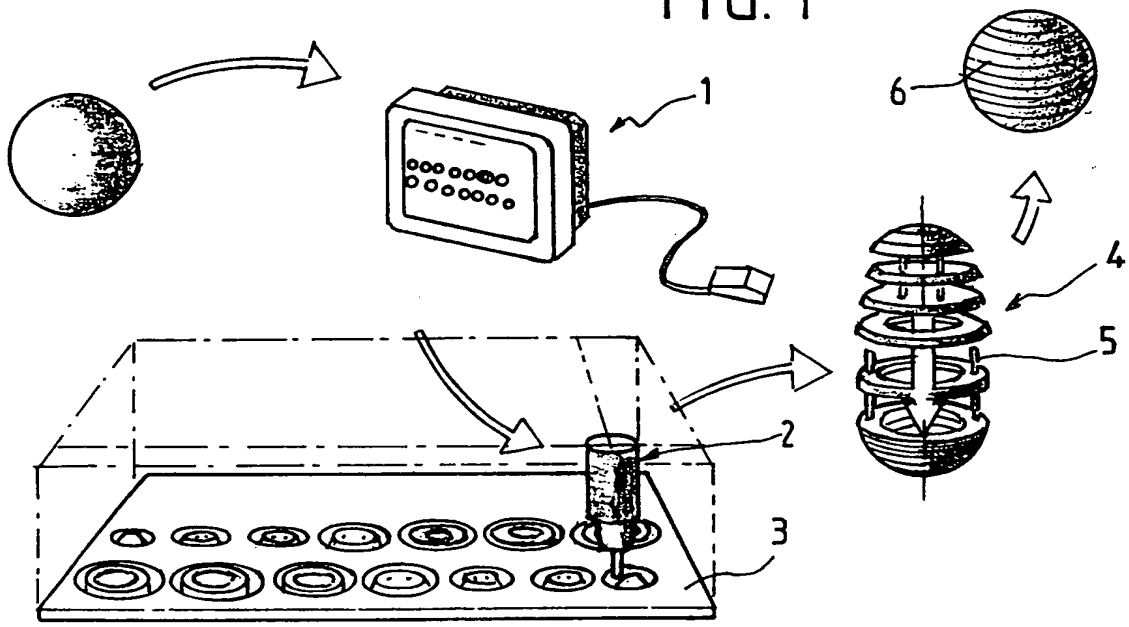


FIG. 2

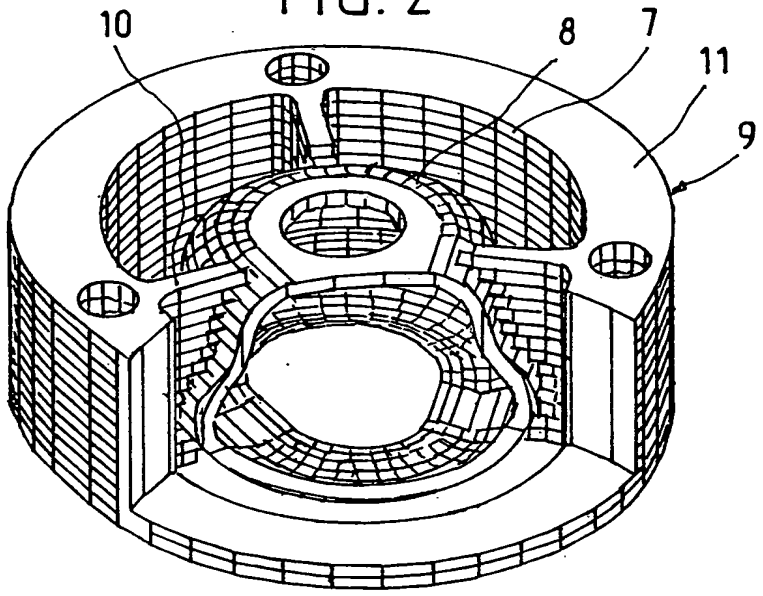


FIG. 3A

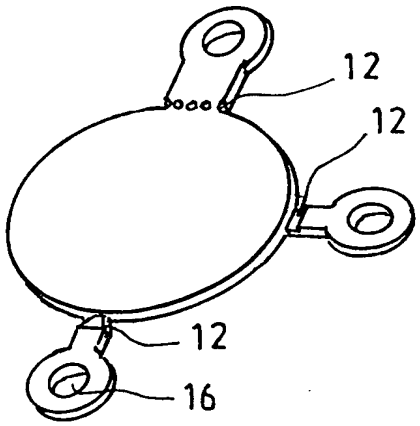


FIG. 3B

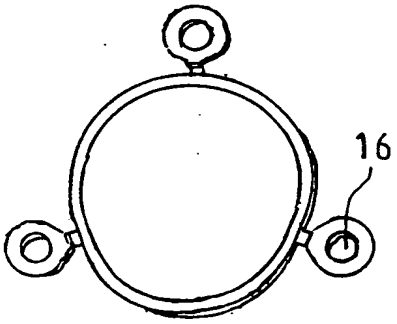


FIG. 3C

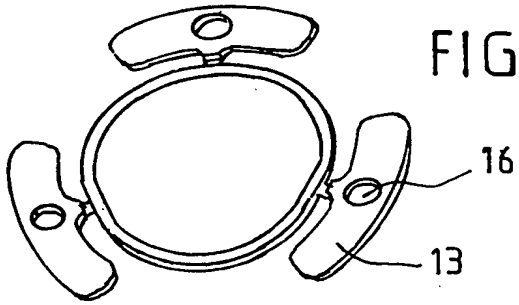


FIG. 4

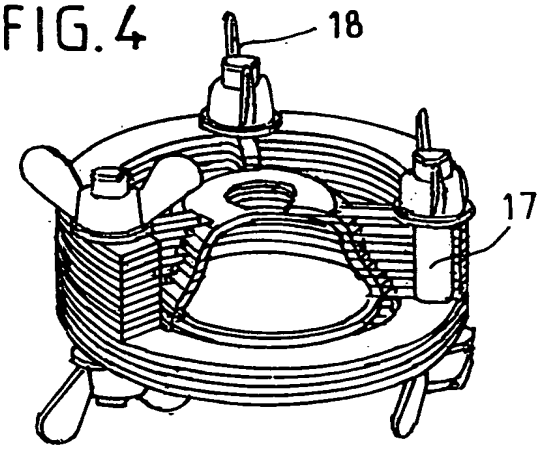


FIG. 6

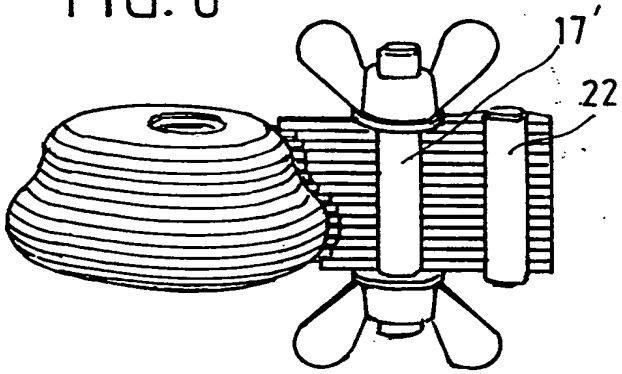


FIG. 5

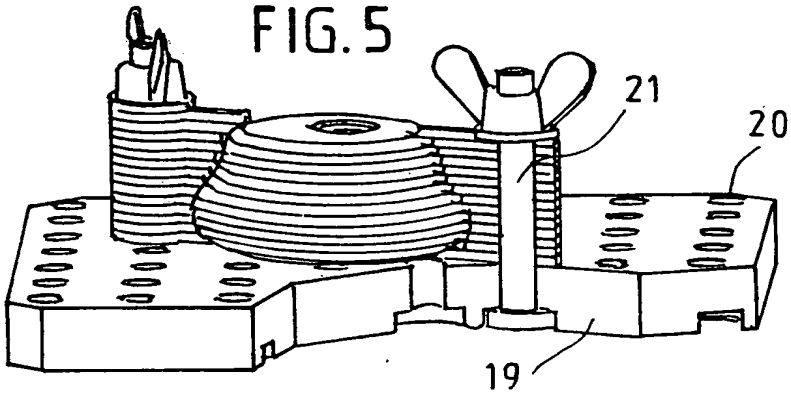


FIG. 3D

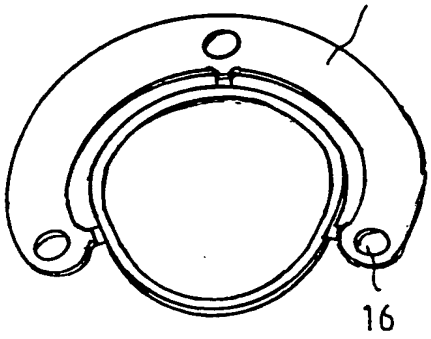


FIG. 7

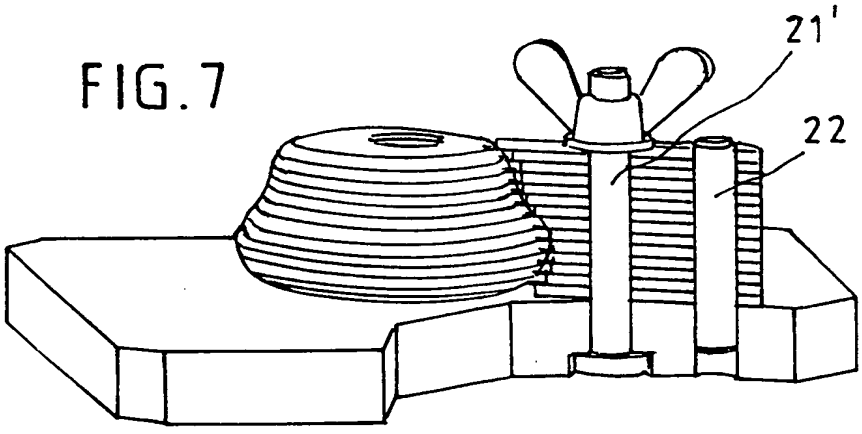


FIG. 3E

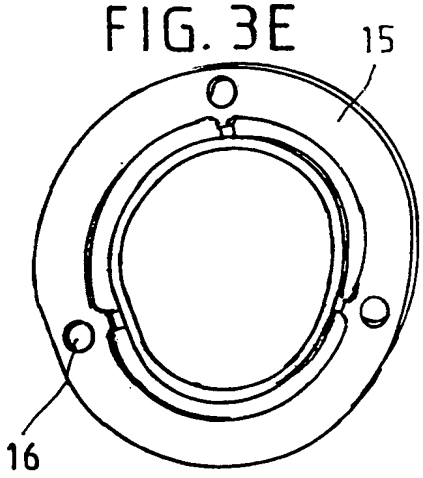


FIG. 8

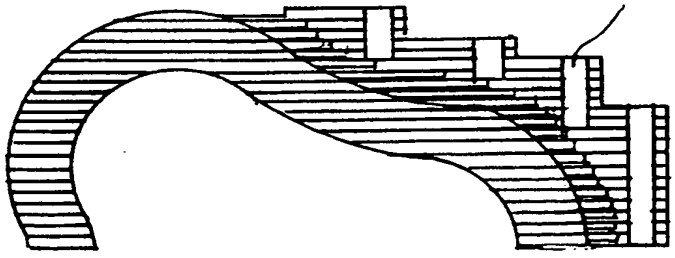
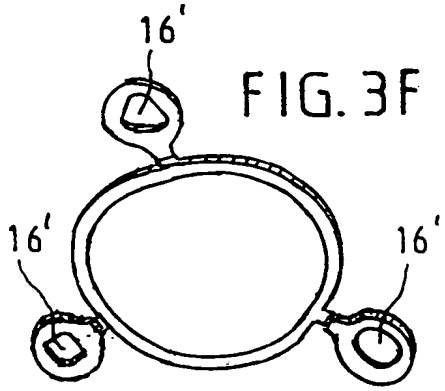


FIG. 3F



M.H

TRAITE DE COOPERATION EN MATIERE DE BREVETS

PCT

RAPPORT DE RECHERCHE INTERNATIONALE

(article 18 et règles 43 et 44 du PCT)

Référence du dossier du déposant ou du mandataire BRL 8 PCT	**POUR SUITE A DONNER** voir la notification de transmission du rapport de recherche internationale (formulaire PCT/ISA/220) et, le cas échéant, le point 5 ci-après	
Demande internationale n° PCT/FR 99/ 02790	Date du dépôt international *(jour/mois/année)* 15/11/1999	(Date de priorité (la plus ancienne) *(jour/mois/année)* 19/11/1998
Déposant C.I.R.T.E.S. et al		

Le présent rapport de recherche internationale, établi par l'administration chargée de la recherche internationale, est transmis au déposant conformément à l'article 18. Une copie en est transmise au Bureau international.

Ce rapport de recherche internationale comprend 3 feuilles.

[X] Il est aussi accompagné d'une copie de chaque document relatif à l'état de la technique qui y est cité.

1. **Base du rapport**

   a. En ce qui concerne la **langue,** la recherche internationale a été effectuée sur la base de la demande internationale dans la langue dans laquelle elle a été déposée, sauf indication contraire donnée sous le même point.

   [ ] la recherche internationale a été effectuée sur la base d'une traduction de la demande internationale remise à l'administration.

   b. En ce qui concerne **les séquences de nucléotides ou d'acides aminés** divulguées dans la demande internationale (le cas échéant), la recherche internationale a été effectuée sur la base du listage des séquences :

   [ ] contenu dans la demande internationale, sous forme écrite.

   [ ] déposée avec la demande internationale, sous forme déchiffrable par ordinateur.

   [ ] remis ultérieurement à l'administration, sous forme écrite.

   [ ] remis ultérieurement à l'administration, sous forme déchiffrable par ordinateur.

   [ ] La déclaration, selon laquelle le listage des séquences présenté par écrit et fourni ultérieurement ne vas pas au-delà de la divulgation faite dans la demande telle que déposée, a été fournie.

   [ ] La déclaration, selon laquelle les informations enregistrées sous forme déchiffrable par ordinateur sont identiques à celles du listage des séquences présenté par écrit, a été fournie.

2. [ ] **Il a été estimé que certaines revendications ne pouvaient pas faire l'objet d'une recherche** (voir le cadre I).

3. [ ] **Il y a absence d'unité de l'invention** (voir le cadre II).

4. En ce qui concerne le **titre,**

   [X] le texte est approuvé tel qu'il a été remis par le déposant.

   [ ] Le texte a été établi par l'administration et a la teneur suivante:

5. En ce qui concerne **l'abrégé,**

   [X] le texte est approuvé tel qu'il a été remis par le déposant

   [ ] le texte (reproduit dans le cadre III) a été établi par l'administration conformément à la règle 38.2b). Le déposant peut présenter des observations à l'administration dans un délai d'un mois à compter de la date d'expédition du présent rapport de recherche internationale.

6. La figure **des dessins** à publier avec l'abrégé est la Figure n° 2

   [X] suggérée par le déposant.

   [ ] parce que le déposant n'a pas suggéré de figure.

   [ ] parce que cette figure caractérise mieux l'invention.

   [ ] Aucune des figures n'est à publier.

INTERNATIONAL SEARCH REPORT

International Application No
PCT/FR 99/02790

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G05B19/4099

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G05B

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 655 317 A (IBM) 31 May 1995 (1995-05-31) the whole document	1,7,8
Y	---	2,4-6
Y	WO 95 08416 A (MASSACHUSETTS INST TECHNOLOGY) 30 March 1995 (1995-03-30) abstract; figures 11,12	2,4-6

X	EP 0 738 583 A (KIRA CORP) 23 October 1996 (1996-10-23) the whole document	1,7,8
Y	---	2,4-6
Y	US 4 001 069 A (DIMATTEO PAUL L) 4 January 1977 (1977-01-04) the whole document	2,4-6

	-/--	

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
7 February 2000	11/02/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Hauser, L

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 585 502 A (ERIN MP EQUIPES DE RECH EN INT) 9 March 1994 (1994-03-09) cited in the application the whole document	1-8
A	EP 0 606 627 A (IBM) 20 July 1994 (1994-07-20) the whole document	1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/FR 99/02790

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0655317 A	31-05-1995	NONE	
WO 9508416 A	30-03-1995	US 5793015 A	11-08-1998
EP 0738583 A	23-10-1996	JP 2905077 B	14-06-1999
		JP 7195531 A	01-08-1995
		JP 2905078 B	14-06-1999
		JP 7195532 A	01-08-1995
		JP 2749504 B	13-05-1998
		JP 7195533 A	01-08-1995
		DE 69414003 D	19-11-1998
		DE 69414003 T	01-04-1999
		EP 0879693 A	25-11-1998
		WO 9518009 A	06-07-1995
US 4001069 A	04-01-1977	US 3932923 A	20-01-1976
EP 0585502 A	09-03-1994	FR 2673302 A	28-08-1992
		AT 145997 T	15-12-1996
		DE 69215673 D	16-01-1997
		DE 69215673 T	19-06-1997
		DK 585502 T	26-05-1997
		ES 2097305 T	01-04-1992
		GR 3022736 T	30-06-1997
EP 0606627 A	20-07-1994	DE 69323858 D	15-04-1999
		DE 69323858 T	05-08-1999
		JP 2558431 B	27-11-1996
		JP 7234888 A	05-09-1995
		US 5587913 A	24-12-1996

RAPPORT DE RECHERCHE INTERNATIONALE

Demande internationale No
PCT/FR 99/02790

A. CLASSEMENT DE L'OBJET DE LA DEMANDE
CIB 7 G05B19/4099

Selon la classification internationale des brevets (CIB) ou à la fois selon la classification nationale et la CIB

B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE

Documentation minimale consultée (système de classification suivi des symboles de classement)
CIB 7 G05B

Documentation consultée autre que la documentation minimale dans la mesure où ces documents relèvent des domaines sur lesquels a porté la recherche

Base de données électronique consultée au cours de la recherche internationale (nom de la base de données, et si réalisable, termes de recherche utilisés)

C. DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie °	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
X	EP 0 655 317 A (IBM) 31 mai 1995 (1995-05-31) le document en entier	1,7,8
Y		2,4-6

Y	WO 95 08416 A (MASSACHUSETTS INST TECHNOLOGY) 30 mars 1995 (1995-03-30) abrégé; figures 11,12	2,4-6

X	EP 0 738 583 A (KIRA CORP) 23 octobre 1996 (1996-10-23) le document en entier	1,7,8
Y		2,4-6

Y	US 4 001 069 A (DIMATTEO PAUL L) 4 janvier 1977 (1977-01-04) le document en entier	2,4-6

	-/--	

[X] Voir la suite du cadre C pour la fin de la liste des documents

[X] Les documents de familles de brevets sont indiqués en annexe

° Catégories spéciales de documents cités:

"A" document définissant l'état général de la technique, non considéré comme particulièrement pertinent

"E" document antérieur, mais publié à la date de dépôt international ou après cette date

"L" document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée)

"O" document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens

"P" document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée

"T" document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention

"X" document particulièrement pertinent; l'inven tion revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément

"Y" document particulièrement pertinent; l'inven tion revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier

"&" document qui fait partie de la même famille de brevets

Date à laquelle la recherche internationale a été effectivement achevée	Date d'expédition du présent rapport de recherche internationale
7 février 2000	11/02/2000

Nom et adresse postale de l'administration chargée de la recherche internationale	Fonctionnaire autorisé
Office Européen des Brevets, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Hauser, L

1

RAPPORT DE RECHERCHE INTERNATIONALE

Internationale No
PCT/FR 99/02790

C.(suite) DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
A	EP 0 585 502 A (ERIN MP EQUIPES DE RECH EN INT) 9 mars 1994 (1994-03-09) cité dans la demande le document en entier	1-8
A	EP 0 606 627 A (IBM) 20 juillet 1994 (1994-07-20) le document en entier	1

Formulaire PCT/ISA/210 (suite de la deuxième feuille) (juillet 1992)

RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements relatifs aux membres de familles de brevets

Demande internationale No
PCT/FR 99/02790

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
EP 0655317 A	31-05-1995	AUCUN	
WO 9508416 A	30-03-1995	US 5793015 A	11-08-1998
EP 0738583 A	23-10-1996	JP 2905077 B	14-06-1999
		JP 7195531 A	01-08-1995
		JP 2905078 B	14-06-1999
		JP 7195532 A	01-08-1995
		JP 2749504 B	13-05-1998
		JP 7195533 A	01-08-1995
		DE 69414003 D	19-11-1998
		DE 69414003 T	01-04-1999
		EP 0879693 A	25-11-1998
		WO 9518009 A	06-07-1995
US 4001069 A	04-01-1977	US 3932923 A	20-01-1976
EP 0585502 A	09-03-1994	FR 2673302 A	28-08-1992
		AT 145997 T	15-12-1996
		DE 69215673 D	16-01-1997
		DE 69215673 T	19-06-1997
		DK 585502 T	26-05-1997
		ES 2097305 T	01-04-1992
		GR 3022736 T	30-06-1997
EP 0606627 A	20-07-1994	DE 69323858 D	15-04-1999
		DE 69323858 T	05-08-1999
		JP 2558431 B	27-11-1996
		JP 7234888 A	05-09-1995
		US 5587913 A	24-12-1996